النمذجه والمحاكاة Simulation and Modelling

- **النموذج Model :**

بنـاء هيكل تصـوري لوصـف النظـام. أو هو تجريد للنظـام يتكون من تجمـع لمعلومـات حول النظام بغرض دراسته.

- **النظام System :**

النظام هو مجموعة من الاشياء تتفاعل وتعتمد على بعضـها البعض. أو هو مجموعة من الكائنات ترتبط مع بعضـها البعض بصورة ما لتحقيق عدد من الاهداف. اذكر امثلـة لبعض النظم ؟

- **بيئة النظام system environment :**

يتأثر النظام بالتغيرات التي تحدث خارجه كما انه يؤثر علي المحيط من حولـه مثل هذه التغيرات تؤثر علي بيئة النظام.

نظام مصنع : وصول الطلبيات arrival orders.

نظام بنك :وصول الزبائن arrival of customers.

- **نموذج النظام model of a system :**

لعمل نموذج للنظام من الضروري فهم النظام وفكرته وحدوده.

مثال:

نظام المصانع (مصنع سيارات) هو عبارة عن مجموعة من الآلآت وقطع الغيار وعمال وهدفه انتاج سيارات.

وغالبا النظام يتأثر بالتغييرات التي تحدث خارجه وتسمي بيئـة النظـام . وفـي نمذجـة النظم من الضـروري تحديـد حـد boundary بين النظـام وبيئتـه وهـذا يعتمـد علـي الغـرض مـن الدراسـة او النموذج.

مثال:

نظام مصنع السيارات يتأثر بوصول الطلبات.

نظام البنك يتأثر بالحد الفاصل لكمية الدخل.

- **مكونات النظام component of a system:**

لفهم وتحليل النظام هنالك عدة مصطلحات يجب ان تُعرف وهي مكونات النظام وهي:

أ- الكيان entity : وهو اهم كائن في النظام مثال الماكينات في المصانع.

ب- الصفات attribute : وهي خصائص الكيان ومكوناته مثال السرعة والسعة.

ج- النشاط activity : وهي الفترة الزمنية لانجاز عمل معين مثال اللحام.

د- حالة النظام state of a system : هي مجموعة من المتغيرات الضرورية لوصف النظام في اي وقت بالنسبة لاهداف النظام مثال حالة الماكينات (تعمل ، لاتعمل ، متعطلة)

هـ- الحدث event : وهو تغبير فوري يغير حالة النظام مثلا توقف النظام، وهنال نوعان من الاحداث هي :

احداث ذاتية النشوء Endogenous : وهي الاحداث التي تؤثر علي النظام.

احداث خارجية Exogenous : وهي الاحداث التي تتكرر مع بيئة النظام.

ملحوظة :

مجموعة الكيانات التي تكون النظام بغرض دراسة واحده يمكن ان تكون مجموعة جزئية لدراسة اخري.

الجدول التالي يوضح المكونات الاساسية لبعض النظم:

Examples of Systems and Components

System	Entities	Attributes	Activities	Events	State Variables
Banking **البنوك**	**Customers** **الزبائن**	**Checking account balance** **التاكد من الرصيد**	**Making deposits** **الايداع**	**Arrival; Departure** **الوصول ، المغادرة**	**Number of busy tellers; number Of customers waiting** **عدد الماكينات المشغولة ، عدد الزبائن المنتظرين**
Production **خط انتاج**	**Machines** **الماكينات**	**Speed; capacity; breakdown rate** **السرعة ، السعة ، معدل التوقف**	**Welding; stamping** **اللحام ، الاختام**	**Breakdown** **التوقف**	**Status of machines (busy, idle, or down)** **حالة الماكينة (مشغولة ، عاطلة ، متوقفة)**
Communications **الاتصالات**	**Messages** **الرسائل**	**Length; destination** **المسافة ، الوجهه**	**Transmitting** **الارسال**	**Arrival at destination** **الوصول في الوجهه**	**Number waiting to be transmitted** **عدد الرسائل في انتظار الارسال**

Table (1): System and component

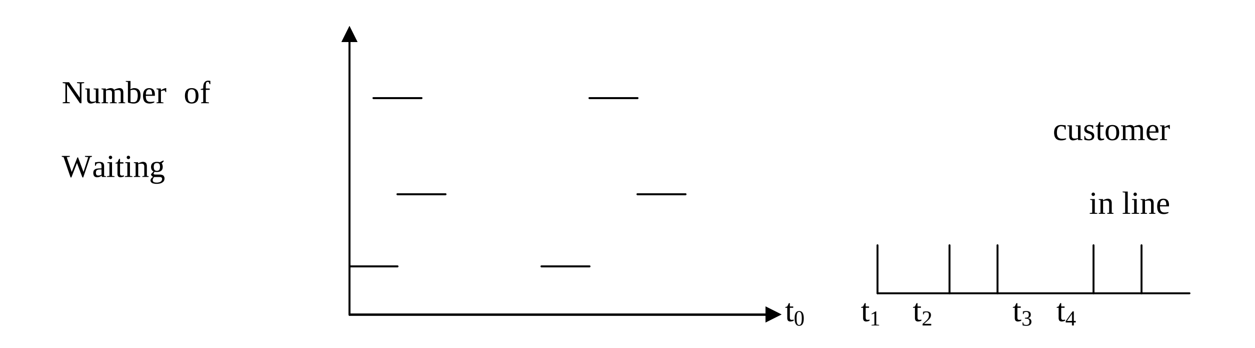
النظم المستمرة والنظم المتقطعة Continuous And Discrete System :

يمكن تقسيم النظم او تصنيفها من خلال تأثرها بالزمن الي نظم مستمرة او نظم متقطعة.

نجد ان هناك القليل من النظم يمكن وصفها بانها متقطعة او مستمرة بصورة دائمة او بشكل كامل ولكن علي حسب اغلب صفات النظام اذا كانت اغلب صفاته متقطعة يوصف بانه متقطع

واذا كانت معظم صفاته مستمرة يوصف النظام بانه مستمر. وفي كلتا الحالتين تحدث التغيرات في النموذج بالنسبة للزمن.

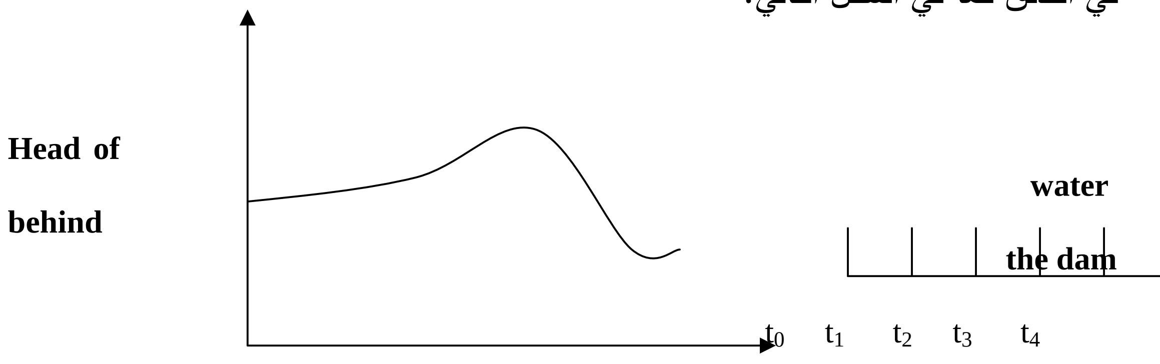
النظم المتقطعة هي التي تتغير فيها حالة المتغيرات او الصفات عند نقاط متقطعة غير مستمرة خلال فترة زمنية مثلا نظام البنك هو نظام متقطع لان حالة المتغير (عدد العملاء) في البنك يتغير عند وصول عميل جديد او عند انتهاء الخدمة للعميل، الشكل التالي يوضح عدد العملاء عند فترة زمنية:



time t

Fig.(1): Discrete System Example

النظم المستمرة هي التي تتغير فيها واحد او اكثر من المتغيرات بصورة مستمرة خلال فترة زمنية مثال طلمبة الماء جانب الخزان خلال فترة المطر (الخريف) كمية الماء تتغير في البحيرة بجانب الخزان والماء يستخدم لعمل الكهرباء والتبخر يقلل من معدل الماء ولكن نجد معدل الماء مستمر في التدفق كما في الشكل التالي:



time t

Fig.(2): Continuous System Example

– نمذجة النظم model of a system :

في بعض الاحيان ندرس النظام لفهم العلاقات بين مكوناته او دراسة كيفية عمل النظام تحت ظروف جديدة، وفي بعض الاوقات من الممكن تجربة النظام نفسه ولكن ليس دائما مثلا

في حالة النظم الجديدة التي لم توجد بعد او التي يمكن ان تكون في شكل افتراضات او في مرحلة التصميم، وحتي اذا كانت النظم موجوده يمكن ان تكون غير مطبقة عمليا لاختبارها.
مثال : تقليص عدد ماكينات tellers لدراسة الاثر في خطوط الانتظار يمكن ان يؤدي الي تقليص العملاء ونقل حسابهم الي بنك اخر.

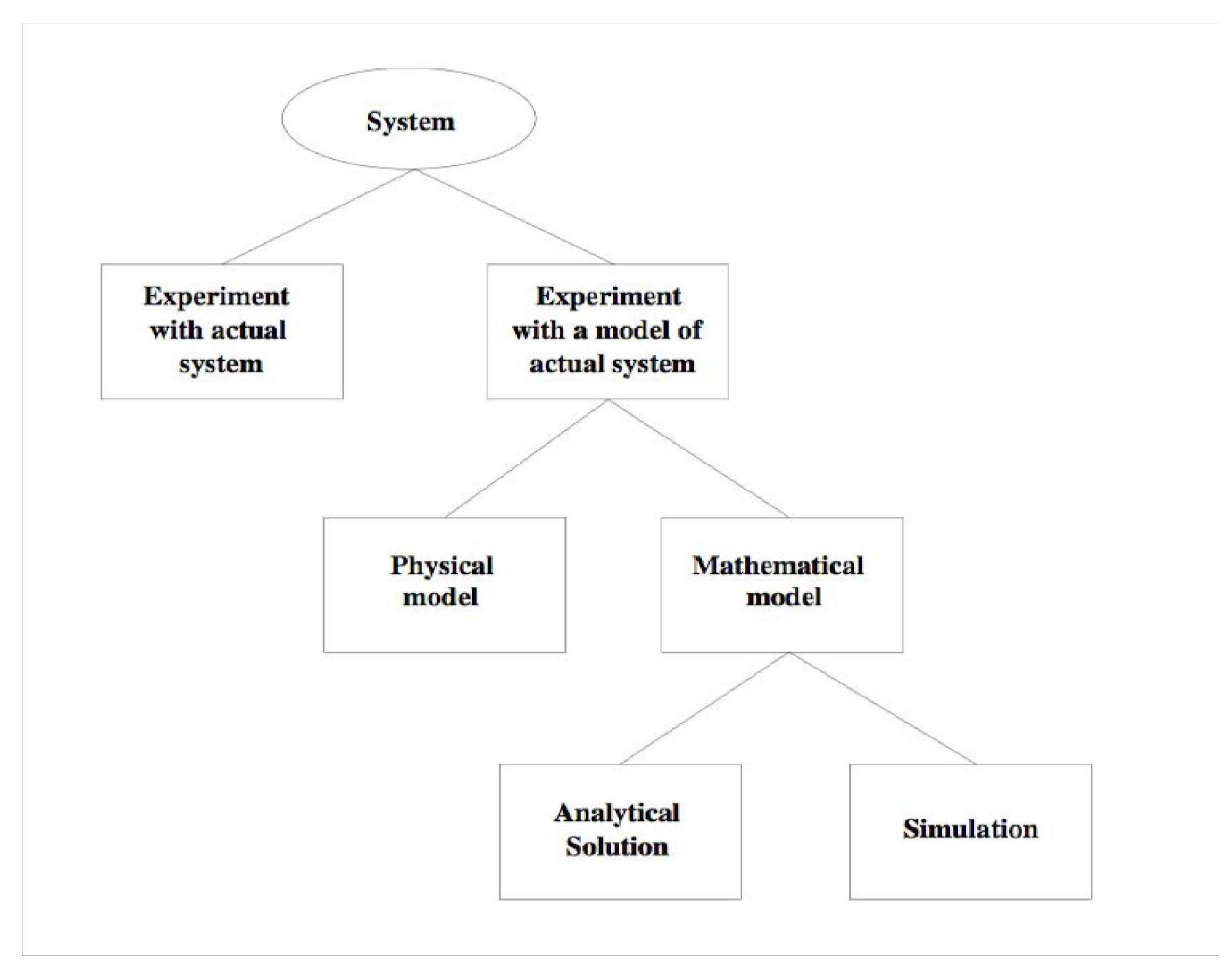
ويعرف النموذج كتمثيل بسيط للنظام بغرض دراسة النظام من اجل فهم النظام بصورة اسهل او التحكم فيه وتحسينه او مراقبة تصرفاته.. في معظم الدراسات من الضروري وضع اعتبارات للتوقعات التي تحدث تأثيرات او تغيرات للنظام وهو في طور المناقشة، وهذه التوقعات تمثل داخل نموذج النظام ، والنموذج علي حسب التعريف هو تبسيط للنظام ، ومن جهه اخري نجد ان النموذج يجب ان يوفر ايجاز صحيح للنظام الحقيقي. والنماذج المختلفة للنظام الواحد يمكن ان تكون مطلوبه لتعبرعن التغييرات في النظام وهو قيد المناقشة.

وعملية النمذجه هي عملية انشاء علاقات متداخلة بين العناصر االاساسية في نظام شامل (الاهداف– الاداء– المعوقات التي تعترض بناء النظام). وعملية النمذجة هي عملية متكرره ومتداخلة وذلك نسبة لاستخدام اكثر من عملية في نظام التغذية العكسية feedback (مرتدة) من نتائج كل عملية لكل مرحلة.

– مكونات النموذج component of a model:

مكونات النموذج هي نفسها مكونات النظام وهي كيانات entities وصفات attributes ونشاطات activities ويمكن ان يحتوي علي اي مكونات اخري تكون مهمة للدراسة.

انواع النماذج types of models :



تقسم النماذج الي :

١- النماذج الفيزيائية (مادية) physical models :

وهي التي تمثل نظم فيزيائية وعملية بنائها مكلفة جدا ،في بعض الاحيان يمكن القول ان النموذج الفيزيائي له وجود مادي ومحسوس ويشغل حيز.

النماذج الفيزيائية تبني بمواد حسية مثل بناء نموذج طائرة في مرحلة التصميم وذلك لاختبار هيكلها تحت ظروف معينه.

تقسم النماذج الفيزيائية الي نماذج ساكنة static وهي التي لاتتغير حالتها مع الزمن (مثل نموذج لبناء المسجد الحرام ونموذج سفينه أو طائره في نافذة مكتب سياحي) ونماذج حركية dynamic وهي التي تتغير حالتها مع الزمن (مثل نموذج آلة الاحتراق الداخلي)

٢- نماذج رياضية (تحليليه) mathematical models :

وهي التي تستخدم الرموز والمعادلات الرياضية لتمثيل النظام ونموذج المحاكاة هو نوع معين من النموذج الرياضي للنظام.

مكونات النماذج الرياضية لاي نظام:

تتكون المكونات الرياضية من اربعة عناصر هي:

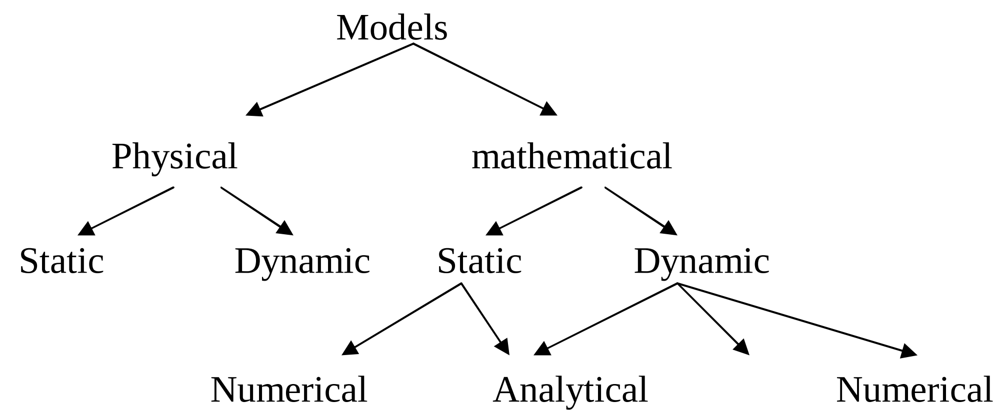
أ- المكونات component.

ب- المتغيرات variables.

ج- العوامل parameters.

د- العلاقات الوظيفية functional relations.

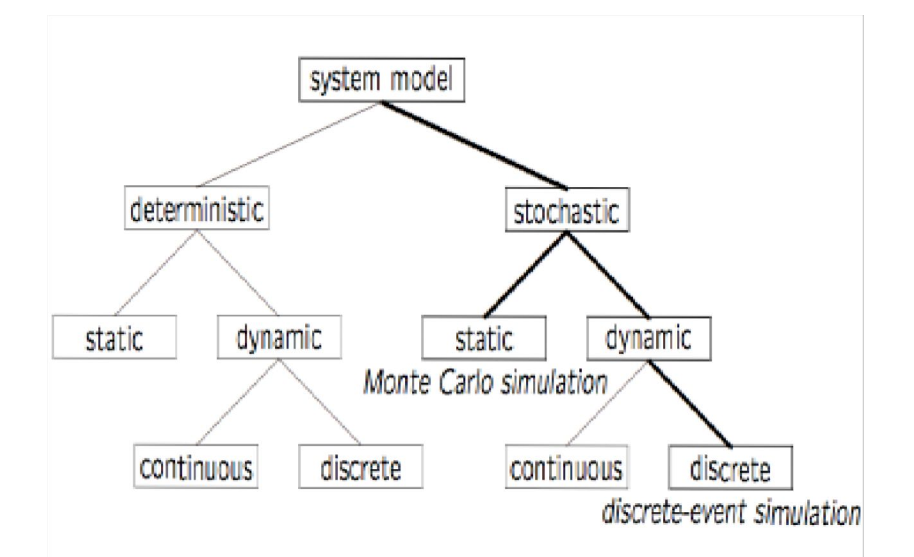
ويمكن تمثيل انواع النماذج بيانيا كالتالي:



Simulation

في التوضيح السابق قسمنا النماذج الرياضية الساكنة الي تحليلية وعددية و النماذج الرياضية الحركية الي تحليلية وعددية ومحاكاة وذلك لأن القليل جدا من النماذج الرياضية يمكن حلها تحليليا زذلك باستخدام النظريات الرياضية والاستنتاجات المنطقية ، واغلبها يمكن حلها عدديا باستخدام الطرق والخوارزميات العددية وجميعها يمكن حلها بواسطة المحاكاة.

تصنيفات النموذج model taxonomy :



خصائص نموذج المحاكاة Characterizing a Simulation Model :

١- نماذج محددة او صدفية deterministic or stochastic models.

النماذج المحددة deterministic هي نماذج المحاكاة التي لا تحتوي علي متغيرات عشوائية وهي التي لها مجموعة واحدة ومحددة من المخرجات.

مثال: جدول مواعيد عند طبيب الاسنان اذا التزم كل مريض بموعده يمكن ان يكون هناك زمن وصول محدد.

النماذج الصدفية stochastic يكون لها واحد او اكثر من المدخلات متغيرات عشوائية والمدخلات العشوائية تؤدي الي مخرجات عشوائية لذلك نستخدم الاحصاء للحصول علي مميزات صحيحة للنظام مثلا متوسط وقت الانتظار للمستفدين ، متوسط عدد الاشخاص المنتظرين.

٢- نماذج ثابتة او متغيرة static or dynamic models.

النماذج الثابتة تسمي محاكاة مونت كارلو Monte Carlo simulation وهي تمثيل للنظام في نقطة زمنية محددة.

النماذج المتغيرة وهي تمثل النظم التي تتغير خلال فترة زمنية محددة مثال: محاكاة نظام بنك من 9:00 am الي 4:00 pm.

٣- نماذج مستمرة او متقطعة continuous or discrete models.

ليس بالضروري استخدام محاكاه النماذج المتقطعة في النظم المتقطعة كما ليس بالضروري استخدم محاكاة النظم المستمرة في النظم المستمرة، ويمكن لنماذج المحاكاة ان تستخدم المتقطعة والمستمرة معا لاختبار اي نموذج مستمر او متقطع وكيفية الاختيار تكون واحدة من اهداف تحديد مميزات النظام واهداف الدراسة.

مثال: في حالة قناة الاتصال يمكن ان نستخدم نموذج متقطع اذا كانت حركة كل رسالة ضرورية للدراسة.

– تطوير النموذج How a model can be developed :

تطوير نموذج المحاكاة يتم بواسطة دراسة كيفية عمل النظام وتطويره خلال فترة زمنية معينة، وهذا النموذج ياخذ عادة شكل مجموعة من الافتراضات متعلقة بتشغيل النظام ، ويعبر عن هذه الافتراضات بعلاقات رياضية او منطقية او رمزية لمكونات النظام (الكيانات Entities او الكائنات object)، وبعد تطوير وتصحيح هذا النموذج يمكن ان يستخدم للاجابة علي استفسارات كثيرة في شكل if .. .. what (ماذا سيحدث ... اذا).

الطرق الرياضية المستخدمة في عمليات تطوير النموذج تشمل نظريات الاحتمالات أو طرق جبرية. مميزات هذه الطرق اذا كانت تستخدم في الانظمة العادية (غير معقده complex systems) تؤدي الى نتائج دقيقه بالرغم من استخدامها لعدد قليل من العوامل.

يمكن كذلك استخدام طرق رياضية اعتمادا علي جهاز الحاسوب computer-based تتميز هذه الطرق بأنها بسيطه ومرنة في حالة الأنظمة المعقدة.

مثال:

تطوير نموذج محاكاة الأحداث المتقطعة Discrete-event simulation DES:

- تحديد الأهداف determine the goals and objectives.
- بناء نموذج تخيلي build a conceptual model.
- التحويل الي نموذج وصفي convert into a specification model.
- التحويل الي نموذج حسابي convert into a computational model.
- التحقق verify.
- الصلاحية validate.

ثلاث مستويات للنموذج Three model levels:

١– نموذج تخيلي conceptual model:

- مستوي عالي جدا
- كم يجب ان يشتمل النموذج
- ماهي المتغيرات الثابتة ، ايها المتغيرة ، ايها المهمة

٢– نموذج وصفي specification model:

- كتابة علي الورق.
- قد يحوي معادلات ، شفرات مزيفة pseudo code.
- كيف يمكن للنموذج ادخال البيانات

٣- نموذج حسابي computational model:

- برنامج حاسوبي.
- لغة عامة او لغة محاكاة.

مقارنة بين التحقق والصلاحية :

التحقق verification:

- النموذج الرياضي يجب ان يكون متطابق consistent مع النموذج الوصفي.
- هل تم بناء النموذج الصحيح؟

الصلاحية validation:

- النموذج الحسابي يجب أن يكون متطابق consistent مع النظام المحلل.
- هل تم بناء النموذج الصحيح؟
- هل يمكن لخبير ان يميز ناتج المحاكاة من ناتج النظام؟

فوائد النماذج :

- تكون تكلفة تحليل النماذج اقل بكثير من تكلفة التجارب الشبيهة التي تجرى على النظام الحقيقي.
- تمكن النماذج من ضغط الوقت.
- تكون معالجة النموذج أسهل كثيرا من معالجة النظام الحقيقي.
- تكون تكلفة حدوث الخطأ في تجربة المحاولة اقل عند استخدام النماذج عما في الواقع.
- يسمح استخدام النماذج بحساب المخاطر في إجراءات محددة.
- تمكن النماذج الرياضية من تحليل عدد كبير من الحلول.
- تعزز النماذج و تقوى التعليم و التدريب.

استخدام النماذج :

- تسهيل الفهم: يكون النموذج ابسط من النظام الذي يفهم بسهولة اكبر عند تمثيل عناصره و العلاقات بينها بطريقة مبسطة.
- تسهيل الاتصال: بمجرد فهم القائم بحل المشكلة للنظام غالبا ما يلزم توصيل هذا الفهم الي آخرين.
- التنبؤ بالمستقبل: يمكن للنموذج الرياضي ان يتنبأ بما يمكن ان يحدث في المستقبل إلا انه قد لا يكون ١٠٠% دقيقا في هذه الحالة.
- يسمى نشاط استخدام النموذج محاكاة

- **المحاكاة Simulation :**

مصطلح لاتيني يعني نسخة او صورة انعكاسية مصغرة والنمذجة باسلوب المحاكاة هي محاولة يتم من خلالها ايجاد صورة انعكاسية مصغرة طبق الاصل لنظام ما دون محاولة الحصول على النظام الحقيقي نفسه وذلك بتطوير نموذج (model) يمثل النظام موضوع الدراسة حيث يظهر جميع التغيرات الممكنة لحالات النظام ثم وضع المقاييس التي تستخدم في تقدير اداء النظام باجراء تجارب على عينات النظام.

أو هي مجموعة من العمليات تحاكي العمليات في العالم الحقيقي او النظم الموجودة خلال فترة معينة سواء كانت تلك النظم يدوية او حاسوبية.

وتستلزم المحاكاة دراسة النظام وملاحظته لدراسة الاثر المتعلق بخصائص تشغيل النظام في العالم الحقيقي.

يمكن استخدام المحاكاة لدراسة النظم في مرحلة التصميم قبل بناء النظام او في مرحلة التحليل كاداة تحليل للتنبؤ بتأثير تغييرات النظم الموجودة او كاداة تصميم للتنبؤ بكفاءة النظم الجديدة عن طريق مجموعة متغيرة من الحالات. وكذلك يمكن محاكاة تغييرات محتملة للتنبؤ بمدي تأثيرها علي اداء النظام.

- المحاكاة باستخدام الحاسوب :

في بعض الامثلة يمكن للنموذج ان يتطور حتي يكون سهل لكي يُحل بواسطة طرق رياضية مثل المعادلات التفاضلية، نظرية الاحتمالات ، الطرق الجبرية او اي تقنيات رياضية اخري. والحل عادة يحتوي علي مجموعة من العوامل العددية تسمي مقاييس كفاءة النظام.

ومع ذلك نجد ان معظم النظم الحقيقية تكون معقدة حتي انه عمل نماذج لحلها رياضيا يكون شبه مستحيل وفي هذه الحالات نستخدم المحاكاة المؤسسه علي الحاسوب. وهذا النوع من المحاكاة يستخدم لمحاكاة تصرفات النظام خلال فترة زمنية معينه من المحاكاة، وتجمع البيانات كما في النظم الحقيقية والبيانات المنتجة من المحاكاة تستخدم لتطوير مقياس كفاءة النظام.

استخدام النماذج الحاسوبية لدراسة التجارب المتعلقة بالأنظمة الحقيقية تم تعريفها من قبل العلماء بعدة طرق نستنتج منها : (المحاكاة هي اسلوب عملي لحل المشاكل المتعلقة بالأنظمة الحقيقية من خلال تصميم نظام يحاكي النظام الحقيقي ومن ثم تصميم نماذج حاسوبية لمعرفة التغيرات الإحصائية والديناميكية للنظام الحقيقي من خلال اختبار النظام الذي يحاكيه).

- نموذج المحاكاة Simulation model :

يمكن وصف ودراسة سلوك الانظمة الموجودة خلال فترة معينة باستخدام نموذج المحاكاة التي تعتبر اسلوب عملي لحل المشاكل المتعلقة بالانظمة الحقيقية من خلال تصميم نظام يحاكي النظام الحقيقي وعليه فان بناء نموذج محاكاة يهتم بالآتي:

١- شرح سلوك النظام.

٢- استخلاص النظريات أو الفرضيات التي تعد لملاحظة سلوك النظام.

٣- استخدام هذه النظريات لتقدير أو تخمين السلوك المستقبلي للنظام (التغيرات الديناميكية).

أهداف المحاكاة simulation objectives :

١. دراسة النظام الحالي.
٢. تحليل بعض الانظمة المقترحة.
٣. تخطيط وتصميم أنظمة اكثر تطوراً.

متي تكون المحاكاة اداة مناسبة When Simulation Is Appropriate Tool:

من العوامل التي جعلت المحاكاة من اكثر الادوات انتشارا وقبولا في بحوث العمليات وتحليل النظم:

- توفراللغات الخاصة بالمحاكاة،
- وانتشار الحاسوب والتقدم في طرق المحاكاة.

ونجد ان المحاكاة لها المميزات الاتيه:

١. المحاكاة لها المقدرة علي دراسة كل ما يتعلق بالنظم الجزئية subsystem للانظمة المعقدة.
٢. محاكاة التغييرات المعلوماتية والادارية والبيئية ودراسة ورؤية تأثير البدائل علي النموذج.
٣. محاكاة النظم في طور التصميم تساهم في تقديم اقترحات لتطوير النظام وهو قيد المناقشة.
٤. تغيير قيم مدخلات المحاكاة ورؤية المخرجات لمعرفة المتغيرات الاكثر اهمية وكيفية تفاعل المتغيرات.
٥. استخدام المحاكاة كاداة تعليم لتقوية وتصحيح طرق الحل التحليلية.
٦. استخدام المحاكاة لتجربة تصميم جديد او طرق جديدة قبل التطبيق لمعرفة ما قد سيحدث.
٧. بمحاكاة قدرات مختلفة للالة يمكن ان نحدد احتياجاتها.
٨. تصميم نموذج محاكاة للتدريب يسمح بالتعلم من غير تكلفة.
٩. النظم الحديثة والمصانع تكون معقدة جدا لذلك التفاعل يتم فقط من خلال المحاكاة.

متي تكون المحاكاة غير مناسبة When Simulation Is not Appropriate:

١. اذا كانت المشكلة تتعلق او تحل بواسطة common sense مثل : اذا كانت لدينا محطة خدمة سيارات وكان وصول السيارات بصورة عشوائية بمتوسط ١٠٠ سيارة في الساعة وتتم الخدمة داخل المحطة بمعدل ١٢ سيارة في الساعة، لحساب العدد الادني

للعاملين في المحطة يمكن حسابهم باستخدام العلاقة(٨.٣ =١٢/١٠٠) اي ٩ عمال او اكثر.

٢. اذا كانت المشكلة تحل تحليليا مثال: متوسط زمن الانتظار في المثال السابق يمكن حسابه من رسم منحني curve.

٣. اذا كان حل المشكلة يكون اسهل باستخدام طرق اختبار مباشرة مثال: في مطعم من الاسهل استخدام عامل اضافي اذا لوحظ بطء في الخدمة.

٤. اذا كانت تكلفة المحاكاة تفوق الدخل.

٥. اذا كان الزمن والمصادر resources غير متوفرين مثال: اذا كان الزمن المسموح به اسبوعين والمحاكاة تستغرق شهر، والمحاكاة تحتاج بيانات كبيرة في حالة عدم توفر البيانات مثلا.

٦. المقدرة لتصحيح وتطبيق النموذج اذا لم يوجد زمن كافي او اشخاص غير متوفرين (المدراء ليس لهم سياسات واضحة).

٧. اذا كان النظام معقد جدا او لا يمكن تعريفه مثال السلوك البشري .

مميزات المحاكاة:

١. يمكن اكتشاف طرق او اجراءات او قواعد او قرارات جديدة من غير تعطيل العمليات الجارية في النظام الحقيقي.

٢. يمكن اختبار تصميم الآلآت hardware جديدة او مخرجات فيزيائية من غير استهلاك او تنفيذ فعلي بشراءهم او تركيبهم.

٣. اختبار المرونة باستخدام فرضيات وتكرار بعض الظواهر.

٤. زيادة او نقصان الزمن لتسريع او تقليل عامل خلال تطور النظام.

٥. معرفة نتائج جديدة باستخدام تفاعلات المتغيرات.

٦. معرفة المتغيرات المهمة لفعالية النظام.

٧. حل مشكلة عنق الزجاجة bottleneck.

٨. دراسة المحاكاة يمكن ان تساعد في فهم كيفية عمل النظام اكثر من الافراد.

٩. اجابة اسئلة what ... ... if.

عيوب المحاكاة:

١. بناء النماذج يحتاج لتدريب خاص (التجربة خلال الزمن).

٢. نتائج المحاكاة يمكن ان تكون صعبة التفسير(الترجمة) لان المخرجات غالبا تكون عشوائية لا المدخلات عشوائية.

٣. تصميم نماذج محاكاة وتحليلها يمكن ان يستهلك زمن وتكلفة عالية.

٤. تستخدم المحاكاة في بعض الاحيان اذا كان التحليل ممكن.

وحلت تلك العيوب كالاتي:

١. منتجي برامج المحاكاة وفروا حزم برامج تحتوي علي جزء او كل النماذج التي تحتاج لمدخلات.

٢. معظم منتجي برامج المحاكاة طوروا المقدرة علي تحليل المخرجات مع برامجهم او حزمهم لانجاز التحليل الكامل.

٣. يمكن ان تتم المحاكاة بسهولة هذه الايام وسريعة بالمقارنة مع السابق وستكون اسرع في المستقبل.

مجالات التطبيق area of application :

اهم مجالات التطبيق هي:

١– التطبيق الصناعي Manufacturing application

٢– تصنيع اشباه الموصلات Semiconductor manufacturing

٣– التطبيقات العسكرية Military application

٤– هندسة البناء وادارة المشاريع Construction engineering and project management

٥– التطبيقات المنطقية وتطبيقات النقل والتطبيقات الموزعة Logic, transportation and distribution application

٦– محاكاة عملية العمل Business process simulation

٧– انظمة الانسان Human systems

من ما سبق يمكن حصر مجالات استخدام نماذج المحاكاة فى حل نوعين من المسائل:

١- المسائل النظرية فى مجالات العلوم الرياضية و الفيزيائية و الكيميائية:مسار الجسيمات على المستوى حل المعادلات تحويل المصفوفات حساب مساحة الاشكال المحددة بمنحنيات.

٢- المسائل التطبيقية فى ادارة و تنظيم مختلف مجالات النشاط الانسانى مثل:

١– محاكاة عملية الانتاج التكنولوجية: التخزين: الصفوف

٢– محاكاة الانظمة الاقتصادية: التخطيط و التنظيم الاقتصادى

٣– محاكاة المسائل الاجتماعية: هجرة السكان و مشاكل السلوك

٤– محاكاة انظمة الطب الحيوي: دورات الدم و عمل الدماغ

٥– محاكاة مسائل التحليل التتابعى لانجاز الاستراتجيات و التكتيك الحربى

المراحل الاساسية لتصميم نموذج محاكاة basic stage of simulation model design :

1. تعريف المشكلة: حيث يتم في هـ١هـ المرحلة وصف المشكلة المدروسة وتحديد الهدف بشكل مفصل.
2. اعداد النموذج بشكل مفصل.
3. ترجمة النموذج الي لغة يتقبلها الحاسوب.
4. التاكد من عمل البرنامج على الحاسوب.
5. التثبيت: حيث يتم التاكد فيها من تطابق مواصفات نموذج المحاكاة الواقعي المدروس.
6. التنفيذ: يتم تنفيذ البرنامج للحصول على المعلومات واستخلاص الناتج والتوصيات لحل المشاكل المدروسة.
7. التقييم: يتم فيها معالجة الناتج إحصائياً.
8. القرار: يتم اتخاذ قرارات من اجل تحسين عمل نظام المحاكاة وتصميم انظمة اكثر تطوراً.

أساليب المحاكاة:

هنالك اسلوبان من المحاكاة:

1. طريقة التناظر (**analogue method**).
2. طريقة مونت كارلو (**monte carlo method**).

١. طريقة التناظر **analogue method:**

في هذه الطريقة يتم تحويل المشكلة قيد الدارسة المطلوب محاكاتها الي مناظر لها بحيث تكون معالجته سهلة والمناظر التقليدي الاكثر استخداما هو الدوائر الكهربية وذلك بتحويل المشكله المراد محاكاتها الي دائرة كهربائية مناظرة بعد تغيير معالم وقواعد القرارت فهذه الطريقة لا تتعامل مع النماذج الرياضية لذا تطبق في المشاكل ذات المتغيرات الكثيرة العدد الي الحد الذي يصعب معه الحل بالطرق العادية. هذه الطريقة محددة التطبيق.

٢. طريقة مونت كارلو **monte carlo method:**

تستخدم هذه الطريقة لمعالجة مختلف انواع المسائل التي تتخللها عمليات عشوائية حيث يصعب عمل تجارب طبيعية يصعب حلها بواسطة الاساليب الرياضية.

تعتمد هذه الطريقة على المحاكاة بواسطة اسلوب العينة وذلك بايجاد عينات من مجتمع نظري يحاكي المجتمع الحقيقي بدلا من اخذ العينات من المجتمع الحقيقي نفسه.

الخطوات المتبعة لمحاكاة مونت كارلو:

١. تحديد نوع التوزيع الاحتمالي للمتغير قيد الدراسة.
٢. إيجاد دالة الكثافة الاحتمالية.
٣. انشاء فترة الارقام العشوائية لكل متغير.
٤. تكوين الارقام العشوائية.
٥. اجراء سلسلة من محاولات المحاكاة.

تعتبر طريقة مونت كارلو مناسبة للآتي:

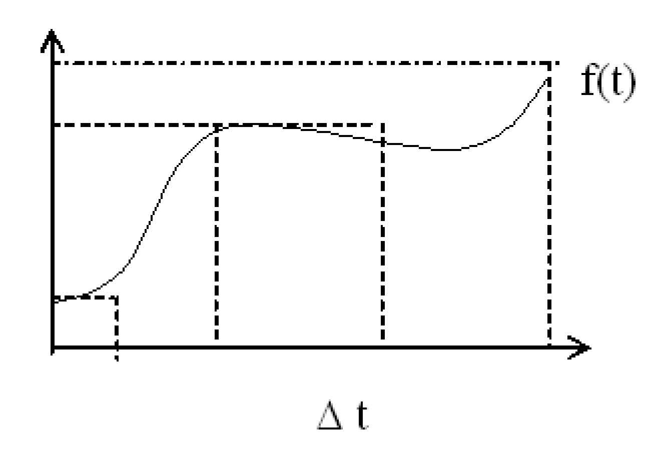
١. تنتج محاكاة مونت كارلو قرارات قيم مدخلات عشوائية اكثر من مدخلات عادية.
٢. توصف المدخلات المتغيرة باحتمالية توزيعاتهم.
٣. توفر وتجهز اداة تحليل حسحسة ومؤثرة.
٤. تسهل اختبار دقة المدخلات.
٥. توفر مجموعة من المخرجات المرئية وذات هدف.

تتضمن هذه الطريقة نوعين من التوزيعات مصنفة وفقا لطبيعة متغيراتها:

١- التوزيع المتصل (Contiguous Distribution)

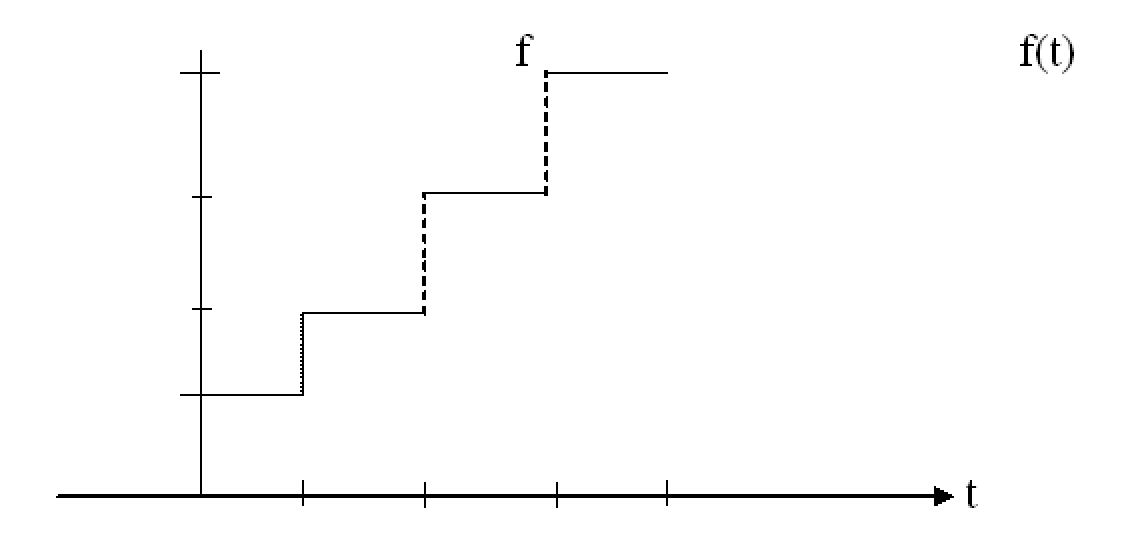
يشمل التوزيع الطبيعي والتوزيع الأسى والتوزيع المنتظم وغيرها بحيث يكون المتغير t محصور بين قيمتين $T \leq t \leq T$

الرسم يوضح دالة توزيع متصلة f(t)



٢- التوزيع المتقطع (Discrete Distribution)

يشمل هذا النوع من التوزيع توزيع برنولى وذي الحدين وبواسونى وغيره . متغير هذا النوع يأخذ قيم محددة حيث تتغير قيمته بمقدارعند نقطة معينة كما يوضحه الرسم التالى :



نجد أكثر أنواع المحاكاة تستخدم للحوادث المتقطعة وذلك لكثرة الظواهر التي تلائمها .

توليد الأرقام العشوائية Generation Of Random Numbers

بما أن محاكاة مونت كارلو تعتمد علي الأرقام العشوائية لذا سنتناول الطرق المختلفة التي يتم بها توليد الأرقام العشوائية والرقم العشوائي هو الرقم الذي يكون احتمال وقوعه مساو لاحتمال وقوع اي رقم عشوائي آخر من مجموعة أرقام عشوائية حيث تتبع الأرقام العشوائية التوزيع المنتظم القياسي [0,1] هذا لان الأرقام العشوائية المولدة بواسطة الآلات الإلكترونية تقع داخل الفترة [0,1] حيث نماذج المحاكاة المعتمدة على الأرقام العشوائية التي تنفذ بواسطة الحاسب .

يتم توليد الأرقام العشوائية عن طريق :

- جداول القيم العشوائية التي يتم تخزينها مباشرة فى ذاكرة الحاسب وهذه طريقة غير مجدية نسبة لحجز حيز كبير فى ذاكرة الحاسب .
- عمليات حسابية تعد لتنفذ بواسطة الحاسب . تتم المفاضلة بين كل عملية واخرى وفقا للاتى :
  - يجب ان تتبع الأرقام العشوائية المولدة التوزيع المنتظم U(0,1) .
  - يجب ان تكون دورة الأرقام العشوائية مستقلة إحصائيا .
  - يجب ان تكون دورة الأرقام العشوائية طويلة .
  - سرعة عملية توليد الأرقام العشوائية

تشتمل طرق العمليات الحسابية على عدة اساليب :

١. طريقة وسط مربع العدد (Mid-Square Method) :
٢. طريقة وسط ضرب العدد (Mid-Product method) :

٣. طريقة باقي القسمة (المطابقة) Congenital :

٤. طريقة المعاينة من التوزيعات الاحتمالية Sampling From Probability Distributions Method:

تشمل طريقة المعاينة من التوزيعات الاحتمالية على طرق تقوم بتوليد عينات عشوائية متتالية (t_1,t_2 …) من توزيع احتمالى f(t)

- كل هذه الطرق أسست على استخدام أرقام عشوائية ذات توزيع منتظم قياسي . Independent and identically distributed uniform (0,1) من هذه الطرق :

١- طريقة المعكوس Inverse Method

٢- طريقة التجميع Convolution Method

طريقة المعكوس Inverse Method

- هي طريقة يتم بموجبها الحصول على متغير عشوائي يتبع توزيعاً معينا لتوليد أرقام عشوائية تتبع ذلك التوزيع وذلك بالاعتماد على الأرقام العشوائية التي تتبع التوزيع الاحتمالي المنتظم القياسي .
- افترض أننا نريد ان نحصل على عينة عشوائية من دالة توزيع احتمالية f(x) سواء كان التوزيع متصل او متقطع . فطريقة المعكوس تقوم أولا بإيجاد دالة الكثافة التراكمية
- $F(x) = P\{ y \le x \}$ حيث $0 \le F(x) \le 1$ لكل قيم y المعرفة ثم نقوم بالخطوات التالية :
- ١- توليد أرقام عشوائية R من التوزيع المنتظم القياس U(0,1) .
- ٢- حساب او أيجاد قيمة x المرادة من $x = F^{-1}(R)$

خوارزمية إيجاد قيم لتوزيع احتمالي متقطع باستعمال طريقة المعكوس:

- إذا كان x يأخذ القيم $x_1, \dots, x_n$ باحتمال $p_1, \dots, p_n$ بحيث

$$\sum_{i=1}^{n} p_i = 1 \qquad 1 \le p_i \le 0$$

- نقوم بالخطوات التالية لإيجاد القيم العشوائية :
- ١- نوجد الدالة التراكمية F_n

$$F_1 = p_1$$
$$F_2 = p_1 + p_2$$
$$\vdots \qquad \vdots$$
$$F_k = \sum_{i=1}^{k} p_i$$
$$\vdots \qquad \vdots$$
$$F_n = \sum_{i=1}^{n} p_i$$

- ٢- نختار عدد عشوائي R_i
- فإذا كان

$$\sum_{i=1}^{n-1} p_i < R_i \le \sum_{i=1}^{n} p_i$$

- اي $F_{n-1} < R_i \le F_n$
- فعندئذ الكمية العشوائية تأخذ القيمة x_n
- يوضح كما يلي :

$$\begin{array}{ll} x_1 & if\ \ 0 < R \le p_1 \\ x_2 & if\ \ p_1 < R \le p_1 + p_2 \\ \vdots & \vdots \\ x_k & if\ \ \sum_{i=1}^{k-1} p_i < R \le \sum_{i=1}^{k} p_i \\ \vdots & \vdots \\ x_n & if\ \ \sum_{i=1}^{n-1} p_i < R \le \sum_{i=1}^{n} p_i \end{array}$$

مثال ١:

ليكن لدينا التوزيع المعبر عن الفترة الزمنية الفاصلة بين تعطيل الالات فى مصنع ما كما يلى :

$$\begin{array}{ll} p(t_n) = 0.12 & t_1 = 4 \\ \quad = 0.48 & t_2 = 5 \\ \quad = 0.22 & t_3 = 6 \\ \quad = 0.18 & t_4 = 7 \end{array}$$

$$n = 1,2,3,4$$

أوجد عشرة قيم عشوائية :

الحل:

نوجد الاحتمالات التراكمية F_n (الفترات)

$$\begin{aligned} F_1 &= p_1 = 0.12 \\ F_2 &= F_1 + p_2 = 0.60 \\ F_3 &= F_2 + p_3 = 0.82 \\ F_4 &= F_3 + p_4 = 1 \end{aligned}$$

- نولد (نختار) أرقام عشوائية R_i فكانت الأرقام التالية

0.4764 , 0.8416 , 0.9434 , 0.3420 , 0.6827

0.8521 , 0.1129 , 0.5806 , 0.9285 , 0.6955

- نأخذ رقم من الأرقام العشوائية R_i بصورة متتابعة وننظر في اي فترة يقع لكي نحدد القيمة العشوائية المطلوبة .
- نجرى الآتي :

$$R_1 = 0.4764 \qquad \because 0.12 < R_1 \leq 0.6$$

$$F_1 < R_1 \leq F_2$$

- إذن القيمة العشوائية ζ:

$$\zeta_1 = t_2 = 5$$

$$R_2 = 0.8416 \qquad \because 0.82 < R_2 \leq 1$$

$$F_3 < R_2 \leq F_4$$

$$\zeta_2 = t_4 = 7$$

وهكذا فنحصل على الجدول التالي :

i	Ri	Fn-1__Fn	tn	ζn
3	0.9434	.82 __ 1	t4	7
4	.3420	.12 __ .6	t2	5
5	.6827	0.6 __ .82	t3	6
6	.8521	0.8 __ 1	t4	7
7	.1129	0.0 __ .12	t1	4
8	.5806	.12 __ .6	t2	5
9	.9285	.82 __ 1	t4	7
10	.6955	0.6 __ .82	t3	6

محاكاة القرار Decision Simulation

يعتمد نجاح المدراء في أعمالهم على نوعية القرارات التي يتخذونها في مواجهة المشاكل والأحوال المتغيرة التي تعترضهم أثناء قيامهم بواجباتهم.تنقسم حالات اتخاذ القرار إلي ثلاث أنواع :

- **اتخاذ القرار في الظروف اليقينية (Decision Making Under Certainty)** يكون متخذ القرار في هذه الحالة على يقين من نتيجة كل بديل من بدائل القرار ومن الطبيعي ان يختار البديل الذي يزيد من عائداته الي أعلى حد ممكن .
- **اتخاذ القرار تحت ظروف المجازفة (Decision Making Under Risk)** يعرف متخذ القرار فى مثل هذه الحالة احتمالية حدوث كل نتيجة ويحاول ان يحسن وضعه الى أعلى حد ممكن فهي من حالات القرارات الاحتمالية فأما ان تحدث زيادة للقيمة المالية المتوقعة الى الحد الأعلى الممكن او تنقص خسارة الفرصة الى الحد الأدنى .
- **اتخاذ القرار فى ظروف غير يقينية(Decision Making Under uncertainty)** لا يعرف متخذ القرار فى مثل هذه الحالة احتمالات النتائج لكل بديل من بدائل القرار فمن الصعب معرفة احتمالية نجاح حزب سياسي فى الانتخابات بعد ٢٠ سنة من ألان ومن المستحيل فى بعض الأحيان تقدير احتمالية نجاح اي استثمار او إنتاج جديد .

محاكاة القرار في حالة المجازفة Simulation of Decision Under Risk

- تتضمن نظم القرارات الإدارية فى حالات المجازفة عناصر احتمالية فى سلوكها
- لذلك يمكننا استخدام طريقة مونت كارلو للمحاكاة.
- لكى نقوم بتوضيح ذلك سنبدأ بمثال لنظام يعتمد على متغير واحد

مثال:

كان الطلب اليومي على قطع السيارات فى أخر ٥٠٠ يوم كما هو مبين في الجدول(١) من معلومات حصلنا عليها من المسئول

الطلب اليومي على السيارات	التكرار

٠	٤٠
١	٨٠
٢	١٠٠
٣	١٢٠
٤	١٠٠
٥	٦٠

الجدول (١)

أجرى عملية المحاكاة للطلب اليومي على قطع السيارات

الحل:

باستخدام طريقة مونت كارلو يمكننا بناء محاكاة للطلب اليومي على قطع السيارات متبعين الخطوات التالية :

١. تحديد الهدف (Define Objective)
   - معرفة الطلب اليومي المتوقع

٢. –تصميم النموذج (Formulating Model) .
   - الطلب اليومي المتوقع = مجموع الطلب اليومي **مقسوم على** عدد الأيام

   (عدد الأيام نقصد به عدد الأيام التي تم فيها إجراء المحاكاة)

٣. تصميم التجربة (Experiment Design)

- بما ان السيطرة على المخزون تعتمد على متغير عشوائي هو الطلب اليومي لذلك نقوم بإنشاء فترة أرقام عشوائية له حيث نتبع الخطوات التالية :

أ- أيجاد الدالة الاحتمالية للطلب اليومي من الجدول (١) .

$$\mathbf{P_i = f_i / \sum f_i}$$

- حيث تشير f_i إلى التكرار المقابل لكل طلب يومي خلال ٥٠٠ يوم والـ p_i دالة التوزيع الاحتمالي للمتغير رقم i.

ب-أيجاد الدالة التراكمية F_n(من الخطوة أ)

$$F_n = \sum_{i=1}^{n} p_i$$

ت-إنشاء فترة I_n للأرقام العشوائية (من الخطوة ب)

$$I_n = F_{n-1} - F_n$$

- مما سبق نحصل على جدول (٢) التالي :

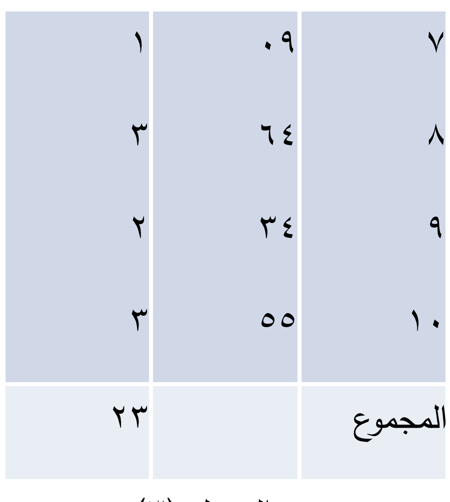
الطلب اليومي	التكرار	الدالة الاحتمالية	الدالة التراكمية	فترة الأرقام العشوائية
0	٤٠	40/500 =.08	.08	١8__1
1	٨٠	80/500 =.16	.24	24__9
2	١٠٠	100/500 =.2	.44	44__25
3	١٢٠	120/500 =.24	.68	68__45
4		100/500=.2	.88	88__69
5		60/500 =.12	1	00__89
	٥٠٠			∑

الجدول (٢)

٤. إجراء التجربة

نقوم باستخدام جدول الأرقام العشوائية لإجراء التجربة ومن خلال الرقم العشوائي يتم تحديد الكمية المطلوبة وفقا للفترة التي يقع فيها الرقم العشوائي المقابلة له من جدول (٢) نتحصل على جدول(٣) التالي :

رقم اليوم	الرقم العشوائي	الطلب اليومي
١	٣٢	٢
٢	٧٣	٤
٣	٤١	٢
٤	٣٨	٢
٥	٧٣	٤
٦	٠١	٠



٧	٠٩	١
٨	٦٤	٣
٩	٣٤	٢
١٠	٥٥	٣
المجموع		٢٣

الجدول (٣)

- تصميم وتحليل النتائج :
- بما ان مجموع الطلب اليومي من جدول(٣) هو ٢٣ عليه يكون معدل الطلب اليومي وفقا للنموذج السابق :

٢.٣= ١٠ \23

أي بمعدل ٢.٣ في اليوم

تمرين

- اذا كان الطلب اليومي على منتج معين يتبع التوزيع الاحتمالي التالي:

الطلب اليومي	الدالة الاحتمالية
٠	٠.٣٣
١	٠.٢٥
٢	٠.٢٠
٣	٠.١٢
٤	٠.١٠

- أجرى عملية المحاكاة للطلب اليومي على المنتج(لمدة ١٥ يوم)

أسلوب المحاكاة للسيطرة على المخزون:

سنحاول في هذا الجزء ان نقدم مشكلة مخزون بمتغير :

تتم السيطرة على المخزون اعتمادا على معلومات سابقة للمدراء بالحصول على مشاهدات عشوائية للسيطرة على المخزون بعد تحديد التوزيع الاحتمالي الملائم وكذلك الحصول على مشاهدات عشوائية لفترة الانتظار (نقطة إعادة الطلب) باستخدام توزيعها الاحتمالي .

بالاستفادة من قيم هذه المشاهدات لكل من الطلب وفترة الانتظار يمكن تحديد مواعيد وضع الطلبية ومواعيد وصولها وكميتها لفترة زمنية قادمة مناسبة وفق احتياجات المنشأة وبالتالي يمكن تقدير التكلفة الإجمالية للتخزين لكل وحدة زمنية متضمنه :

- تكاليف الطلب .
- تكاليف التخزين .
- نفاد المواد المخزونة .

لتوضيح ذلك نتناول المثال التالي الذي يعتمد على متغيرين .

مثال:

يقوم مالك مستودع ببيع نوع معين من المكانس الكهربائية . يعتبر الطلب اليومي لهذا النوع من المكانس قليل نسبيا ولكن معرض لبعض التغيرات لاحظ مالك المستودع خلال ٥٠٠ يوم حجم المبيعات كما فى الجدول (١) التالي :

الطلب اليومي	التكرار
٠	٣٠
١	٦٠
٢	١٠٠
٣	٢٠٠
٤	٧٥
٥	٣٥

الجدول (١)

كما أن هنالك تأخر فى الاستلام هذا من ملاحظته لاستلام ٨٠ طلب وكان الوقت التمهيدي لإعادة الطلب كما فى الجدول (٢) التالي :

الوقت التمهيدي	**التكرار**
١	**٢٠**
٢	**٣٥**
٣	**٢٥**

الجدول (٢)

الحل:

باستخدام طريقة مونت كارلو يمكننا بناء محاكاة للسيطرة على المخزون متبعين الخطوات التالية :

١. **تحديد الهدف : (Define Objective)**

حساب التكلفة الإجمالية للتخزين .

٢. **تصميم النموذج : (Formulating Model)**

بما أن التكلفة الإجمالية للتخزين تعتمد على معدل المخزون اليومي وعدد الطلبات ومعدل الخسارة للمبيعات عليه يكون النموذج :

- **التكاليف الكلية للتخزين =**

(تكاليف إيداع طلب واحد) (معدل عدد الطلبات) +

(تكاليف تخزين كل مكنسة) (معدل المخزون اليومي)+

(تكاليف كل بيع خاسر) (معدل خسارة المبيعات)

٣. **تصميم التجربة (Experiment Design) :**

بما أن السيطرة على المخزون تعتمد على متغيرين عشوائيين هما الطلب اليومي والوقت التمهيدي لذلك نقوم بإنشاء فترة أرقام عشوائية لهما حيث نتبع الخطوات التالية :

أ/ إيجاد الدالة الاحتمالية للطلب اليومي من الجدول (١) .

$$P_i = f_i / \sum f_i$$

ب/ إيجاد الدالة التراكمية F_n(من الخطوة أ)

$$F_n = \sum_{i=1}^{n} p_i$$

ج/إنشاء فترة I_n للأرقام العشوائية (من الخطوة ب)

$$I_n = F_{n-1} - F_n$$

مما سبق نتحصل على جدول (٣) التالي :

الطلب اليومي	التكرار	الدالة الاحتمالية	الدالة التراكمية	فترة الارقام العشوائية

٠	٣٠	30/500 =.06	.06	6__1
١	٦٠	60/500 =.12	.18	18__7
٢	١٠٠	100/500 =.2	.38	38__19
٣	٢٠٠	200/500 =.4	.78	78__39
٤	٧٥	75/500 =.15	.93	93__79
٥	٣٥	35/500 =.07	1	00__94

الجدول (٣)

نستخدم الأجراء السابق للجدول (٢) الخاص بالوقت التمهيدي نحصل على :

الطلب اليومي	التكرار	الدالة الاحتمالية	الدالة التراكمية	فترة الارقام العشوائية
١	٢٠	20/80 =.25	.25	١-٢٥
٢	٣٥	35/80 =.44	.69	٢٦-٦٩
٣	٢٥	25/80 =.31	1	٧٠-٠٠
المجموع	٨٠			

الجدول (٤)

٤-إجراء التجربة :

نقوم باستخدام جدول الأرقام العشوائية لإجراء التجربة ومن خلال الرقم العشوائي يتم تحديد الكمية المطلوبة والوقت التمهيدي إذا افترضنا أن المخزون كان به ١٢ مكنسة كهربائية ونقطة إعادة الطلب ٦ وحدات (إذا كان المخزن به ٦ وحدات او اقل يطلب ١٢ مكنسة) نحصل على الجدول التالي :

اليوم	الوحدات المستلمة	المخزون الاولي	الرقم العشوائي للطلبية	الطلب	المخزون النهائي	المبيعات الخاسرة	هل يطلب	رقم عشوائي للوقت	الوقت التمهيدي
١	-	١٢	٧	١	11	-	No	-	-
٢	-	١١	٦٠	٣	8	-	No	-	-
٣	-	٨	٧٧	٣	5	-	Yes	٩٤	2
٤	-	٥	٧٦	٣	2	-	No	-	-
٥	-	٢	٩٥	٥	0	-3	No	-	-
٦	12	١٢	٥١	٣	9	-	No	-	-
٧	-	٩	١٦	١	8	-	No	-	-
٨	-	٨	١٤	١	7	-	No	-	-
٩	-	٧	٨٥	٤	3	-	Yes	٩٥	2
١٠	-	٣	٨٥	٤	0	-1	No	-	-
المجموع					٥٤	٤		٢	

الجدول (٥)

- تمت معالجة أعمدة الجدول كالآتي :

- **العمود الأول** يعنى تكرار التجربة ١٠ مرات حيث يمكننا تكرارها أكثر من ذلك .العمود الثاني (الوحدات المستلمة) لا يتم استلام أي عدد من الوحدات سوي ١٢ وحدة هذا فى حالة وصول المخزون النهائى ٦ وحدات او اقل وبعد انقضاء الوقت التمهيدي .
- **العمود الثالث** (المخزون الاولى) بدأ كما اشرنا سابقا بـ١٢ وحدة ثم يأخذ اليوم التالي قيمة المخزون النهائى لليوم الحالي الى ان تاتى نقطة إعادة الطلب هذا يعنى ان المخزون الاولى لليوم التالي سيكون ١٢ وحدة مضافة الى المخزون النهائى لليوم الحالي .
- **العمود الرابع** (الرقم العشوائي للطلبية) توجد به الأرقام العشوائية فى صورة متسلسلة تم اختيارها من جدول الأرقام العشوائية باستعمال عمود معين .
- **العمود الخامس** (الطلب) يتم ملأه من خلال العمود السابق وذلك بعد معرفة الفترة التي يقع فيها الرقم العشوائي من الجدول (٣) ومن ثم تحديد الطلبية المقابلة له .
- **العمود السادس** (المخزون النهائى) عبارة عن حاصل طرح الطلبية من المخزون الاولى لنفس اليوم ويأخذ القيمة صفر إذا كان المخزون الأولى اقل من الطلبية او يساويها .
- **العمود السابع** (المبيعات الخاسرة) لا تكون هنالك مبيوعات خاسره إلا إذا كان المخزون النهائى أقل من الطلبيه فهي عبارة عن الوحدات المكملة للطلبيه التي لم يتمكن من استلامها لنفاد المخزون
- **العمود الثامن** (هل يطلب) يكون الجواب لا إلا إذا كان المخزون النهائى ٦ وحدات أو أقل بعد انقضاء الوقت التمهيدي إذا وجد.

- **العمـود التاسـع** (الـرقم العشـوائي للوقـت) عبـارة عـن تسلسـل أرقـام العمـود الرابـع حيـث لاننلجأ إليه إلا أذا كان هنالك طلبيه.

- **العمود الأخير** (الوقت التمهيدي) يتم تحديده عن طريق العمود السابق وذلك من خلال معرفة الفترة التي يقع فيها الرقم العشوائي فى جدول (٤) لتحديد الوقت التمهيدي المقابل له.

- **تصميم وتحليل النتائج :**

- أ – معدل المخزون النهائي = مجموع المخزون النهائي ***مقسوم على*** عدد الأيام

٥٣\١٠ =٥.٣

ب – معدل الخسارة للمبيعات = مجموع الخسارة ***مقسوم على*** عدد الأيام

٤\١٠ =.٤

- ت – معدل عدد الطلب = مجموع الطلبات ***مقسوم على*** عدد الأيام

٢\١٠ = .٢

- فإذا قدر مالك المستودع من خبرته السابقة أن تكاليف إيداع أي طلب هو ٥ دينار وتكاليف تخزين أي مكنسة نهاية اليوم هو 0.2 دينار وتكاليف كل بيع خاسر هو ٣ دينار وبالتالي تكون تكاليف التخزين اليومي في سياسة الطلب على الكمية المخزنة

$$(0.2)(5) + (5.3)(0.2) + (0.4)(3) = 3.26$$

تمرين

- طلب الشراء اليومي على منتج معين يتبع التوزيع الاحتمالي كما فى الجدول (١) التالي :

الطلب اليومي	الدالة الاحتمالية

.33	٠
.25	١
.20	٢
.12	٣
.10	٤

(١)

- يفحص المخزون من هذا المنتج كل سبعة أيام فإذا وجد انه اقل من او يساوى ٦ وحدات يتم طلب ١٠ وحدات الوقت التمهيدي يتبع التوزيع الاحتمالي كما فى الجدول (٢) التالي :

الدالة الاحتمالية	الوقت التمهيدي
.3	١
.5	٢
.2	٣

(٢)

- اذا كان المخزون الابتدائي ١٢ وحدة اجري محاكاة لمدة ٣ أسابيع.حدد نسبة الأيام التي حدث فيها نقص في تحقيق طلبيه ونسبة الوحدات التي طلبت ولم تحقق ومتوسط عدد الوحدات المطلوبة للشراء أسبوعيا.

steps in a simulation study خطوات دراسة المحاكاة

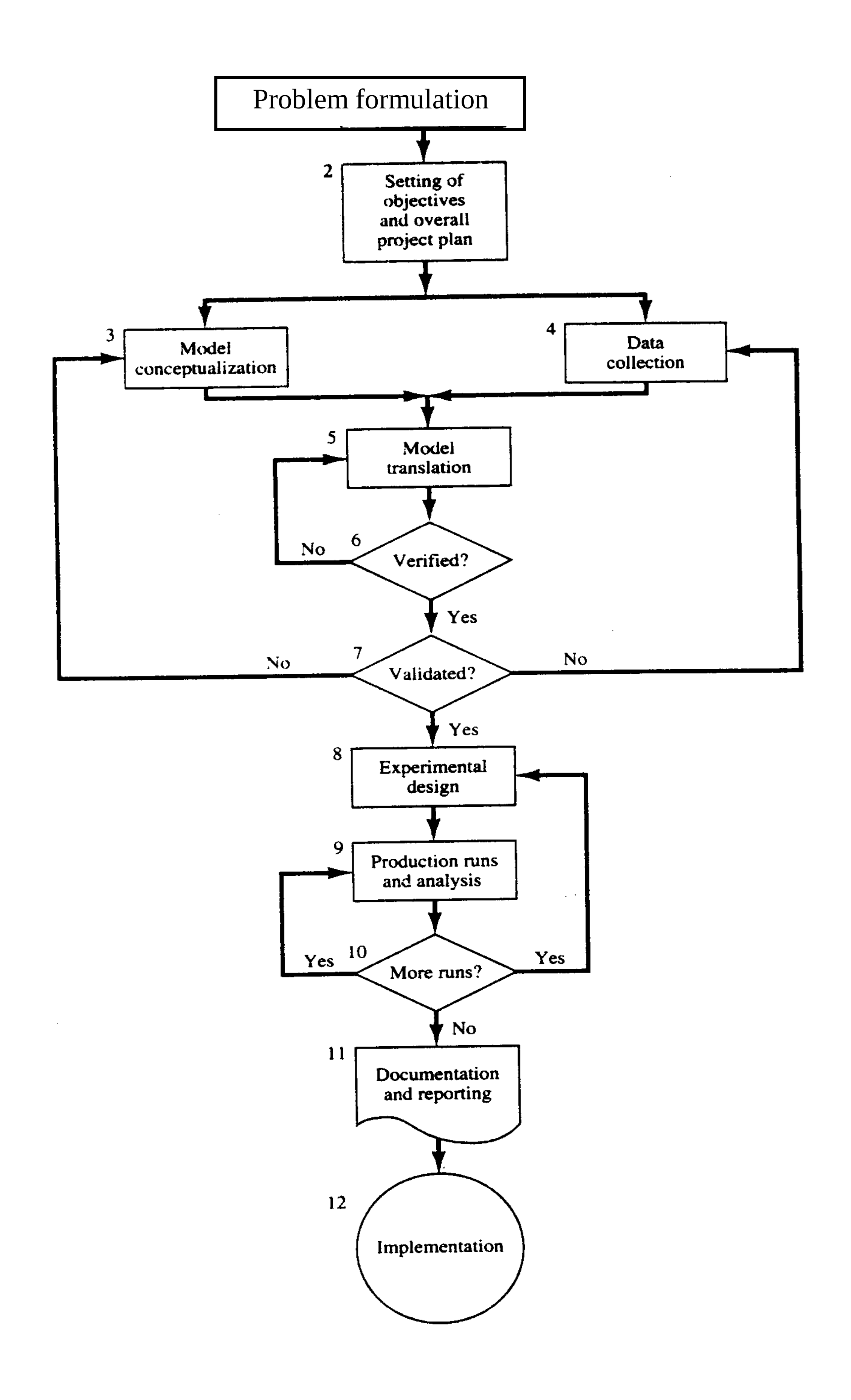


Fig Steps in a simulation study

١. صياغة المشكلة **problem formulation:**

كل دراسة يجب ان تبدأ برؤية وصياغة المشكلة واذا كانت هذه الصياغة تعطي بواسطة صانعي القرار او الذين لديهم المشكلة يجب علي محلل النظام ان يتاكد من ان المشكلة قد وصفت بوضوح وفهم تام، واذا كانت المشكلة قد وصفت بواسطة محلل النظام من الضروري ان يفهم صانعي القرار ويوافقوا علي هذه الصياغة.

٢. وضع الاهداف وخطة المشروع **setting of objective and overall project plan :**

الاهداف تتضمن وتحتوي الاسئلة التي يجب ان تجيب عليها المحاكاة. وفي هذه المرحلة يكون التحديد اذا كانت المحاكاة طريقة مناسبة لصياغة المشكلة وتحديد الاهداف.

واذا قررنا ان المحاكاة مناسبة، يجب ان تحتوي خطة المشروع علي بيانات بالنظم البديلة وطرق تقييم الكفاءة لتلك البدائل ويجب ان تحتوي علي خطط لدراسة عدد الاشخاص المستلزمين لتنفيذ الخطة والتحقق من كفاءة الدراسة والمدة الزمنية (عدد الايام) لانجاز كل خطوة في العمل مع النتائج المتوقعة في نهاية كل مرحلة.

٣. مفهوم النموذج **Model Conceptualization:**

بناء نموذج للنظام هو فن اكثر من اي شئ اخر .بالرغم من انه ليس بالامكان ان نجهز مجموعة من الاوامر تقود لبناء نماذج مناسبة في كل الحالات.ولكن هنالك بعض الارشادات التي يجب ان تتبع. وفن النمذجة هو المقدرة علي تجريد الافتراضات الاساسية التي توضف النظام ومن ثم توسيع وتطوير وزخرفة النموذج حتي نحصل علي النتائج التقديرية المناسبة. ولذلك من الافضل البداية بنموذج بسيط ومن ثم اضافة بعض التعقيدات تدريجيا.

٤. جمع البيانات **data collection:**

هنالك علاقة بين بناء النموذج وجمع بيانات المدخلات التي نحتاج اليها و اضافة اي تعقيدات في صفات النموذج يمكن ان تؤدي الي تغيير عناصر بيانات المدخلات. من الافضل ان نبداء بجمع البيانات مع الخطوات الاولي لبناء النموذج لان جمع البيانات يحتاج الي وقت كبير.

في كثير من الاحيان تحدد الاهداف من الدراسة نوعية البيانات التي تجمع. مثلا في حالة دراسة نظام البنك ، اذا كان الهدف من الدراسة دراسة مدة الانتظار في صف الخدمة لماكينات الصرف الالي نوعية البيانات تكون زمن وصول العملاء وزمن خدمة اله الصرف وزمن الانتظار تحت شروط متغيرة.

٥. ترجمة النموذج **model translation:**

لان عملية تحويل النظم الحقيقة الي نماذج تحتاج الي تخزين معلومات لذلك يجب ان يترجم النموذج ويدخل في شكل حاسوبي ونستخدم التسمية برنامج program حتي نحصل علي صيغة مناسبة للترجمة.

الشخص المسئول من النمذجة (modeler) يجب ان يقرر ما اذا كانت برمجة النموذج تتم بلغات النمذجة العامة مثل GPSS(general purpose simulation software) او استخدام برامج نمذجة خاصة special purpose software.

٦. التحقق من الصحة **?Verified:**

التحقق من الصحة يقصد بها التحقق من صحة برامج الحاسب المجهز لمحاكاة النموذج والاجابة علي السؤال هل البرنامج يعمل بكفاءة ؟. مع النماذج المعقدة يكون من الصعب او المستحيل ترجمة النموذج الي برامج بنجاح من غير خبرة جيدة في العثور علي الاخطاء debugging .

اذا كانت عناصر المدخلات والتمثيل المنطقي للنموذج صحيحين في الحاسوب فبالتالي التحقق من الصحة يكون قد اكتمل. وفي معظم يستخدم الاحيان الاحساس العام (common sense) لانجاز هذه الخطوة.

٧. التصديق الرسمي **validated؟**

الشرعية او الاثبات هي تحديد ما اذا كان هذا النموذج تمثيل دقيق للنظام الحقيقي والشرعية يتوصل اليها دائما من خلال :

١. فحص النموذج
٢. المعالجة البديلة لمقارنة النموذج مع النظام الحقيقي.
٣. استخدام التعارض بين البدائل لتطوير النموذج.

ونكرر هذه المعالجة حتي نحصل علي دقة مقبولة النموذج.

٨. التصميم التجريبي **experimental Design:**

وفي هذه الخطوة يتم تحديد البدائل التي يجب ان تحول باستخدام المحاكاة (تحاكي). وهذا التحديد يتعلق بما اذا كانت هذه البدائل يمكن ان تحقق التشغيل وقد اكتملت وتم تحليلها .ولكل تصميم نظام محاكي القرارات التي نحتاجها هي: لتحديد مدة الانشاء ومدة تنفيذ المحاكاة وعدد النسخ replication لانجاز التنفيذ.

٩. نتائج التنفيذ والتحليل :**Production runs and analysis**

تستخدم نتائج التنفيذ والتحليل لتحديد وتعيين مقياس كفاءة وتقييم النظام الذي صمم له نموذج المحاكاة.

١٠. تشغيل اكثر ؟ **?More Runs**

وهذه الخطوة مؤسسه علي تحليل التشغيل الذي اكتمل ومحلل النظام يحدد اذا كان هنالك احتياج الي تنفيذ اكثر للوصول لكفاءة تشغيل اكثر.

١١. التوثيق والتقرير **documentation and reporting:**

هنالك نوعان من التوثيق هما توثيق البرنامج وتوثيق التطور في البرنامج . ويكون التوثيق بوصف دقيق للنظام وكيفية استخدامه حتي يستطيع كل المتعاملين معه التعامل معه بسهوله وتتضمن هذه الخطوة كتابة تقرير عن النظام ككل.

١٢.التطبيق **implementation:**

ونجاح خطوة التطبيق تعتمد علي نجاح الخطوات السابقة.

المفاهيم الأساسية لمحاكاة النظم المتقطعة

General Principle to Concept in Discrete-Event Simulation

في هذا الجزء نقدم ونشرح المفاهيم الاساسية المستخدمة في اغلب حزم محاكاة النظم المتقطعة وهي لا ترتبط بحزمة معينه.

مفاهيم في محاكاة النظم المتقطعة **Concepts in discrete-event simulation:**

١. النظام **system:** مجموعة من الكائنات ترتبط مع بعضها البعض بصورة ما لتحقيق عدد من الاهداف.

٢. النموذج **model** : هو وصف مبسط للنظام.

٣. حالة النظام **system state** هي مجموعة من المتغيرات الضرورية لوصف النظام في اي وقت بالنسبة لاهداف النظام.

٤. الكيان **Entity** وهو اهم كائن في النظام.

٥. الصفات **Attribute** وهي خصائص الكيان ومكوناته.

١. القائمة **List** هي مجموعة من الكيانات مرتبة بطريقة منطقية.

٢. الحدث **Event** وهو حالة تُحدث تغبير فوري يغير حالة النظام.

٣. ملاحظة الحدث **Event Notice** وهي سجل للحدث ويحتوي عادة علي اسم الحدث وفترته الزمنية (event type, event time).

٤. قائمة الاحداث **Event List** وهي قائمة بالاحداث المستقبلية والاحداث الوشيكة الحدوث وترتب علي حسب وقوعها وتكتب في شكل قائمة تسمي قائمة الاحداث المستقبلية(Future Event List (FEL.

٥. النشاط **Activity** وهي الفترة الزمنية لانجاز عمل معين. وهي مدة من الزمن محددة الطول وتعرف منذ البداية مثل زمن الوصول وزمن الخدمة.

٦. التأخير **Delay** وهي فترة غير محدة الطول .

٧. الساعة **Clock** وهي متغير يمثل زمن المحاكاة.

- تستخدم بعض حزم المحاكاة مصطلحات مختلفة لنفس المفاهيم السابقة. مثلا القائمة List تسمي في بعض الحزم مجموعة Set او صف Queue او سلسلة chain.

- ترتيب الكيانات في قائمة يُرتب دائما بواسطة قاعدة مثل: First In First Out (FIFO) او Last In First Out (LIFO) او ترتب بصفة من الكيان مثل الاولوية او التاريخ
- قائمة الاحداث المستقبلية FEL تُرتب دائما علي حسب زمن الحدث event time المسجل في ملاحظة الحدث Event Notice.
- والنشاط Activity يمثل زمن الخدمة ، زمن الوصول او اي معالجة زمنية اخري لها مده زمنية معرفة بواسطة واضع النموذج modeler. ومدة النشاط يمكن ان تحدد بعدة طرق ويمكن ان تكون:

١. محددة deterministic مثلا تكون دائما ٥ دقائق.

٢. احصائية statistical مثال رقم عشوائي من باحتمالية متساوية.

٣. دالة تعتمد علي متغيرات النظام (او/و) صفات الكيان مثال زمن التحميل لسفينة كدالة لحمولة السفينة ومعدل التحميل كطن في الساعة.

- التأخير delay يسمي في بعض الاحيان بالانتظار الشرطي conditional wait والنشاط يسمي الانتظار غير الشرطيunconditional wait. اكتمال النشاط activity هو حدث event. وغالباً يسمي الحدث الابتدائي primary event ويدار باحلال ملاحظة الحدث event notice في FEL.
- وعلي العكس نجد ان التأخير يدار بواسطة احلال الكيان المساعد associated entity في قائمة اخري او في صف الانتظار. واكتمال التأخير يسمي في بعض الاحيان شرطي conditional او حدث ثانوي secondary event ولكن هذه الاحداث لا تمثل بواسطة event notice ولا تظهر في FEL.
- والانظمة التي تُعالج هنا هي انظمة متحركة وهذه يعني انها تتغير خلال فترة زمنية لذلك حالة النظام، وصفات الكيان وعدد من الكيانات النشطة والنشاطات والتأخير تتغير جميعها خلاة فترة من الزمن.والزمن نفسه يمثل بواسطة متغير يسمي clock.

تنفيذ المحاكاة بالحاسوب Computer Implementation Of Simulation:

الصعوبة المصاحبة لتنفيذ نماذج المحاكاة بالحاسوب باستخدام لغات برمجة ذات اغراض عامة مثل البيسك والباسكال ادت الي اعداد العديد من لغات محاكاة ذات اغراض خاصة وتكون هذه اللغات في بعض الاحيان ذات غرض خاص بدرجة لا تكون مناسبة لمحاكاة نظم اخري.مثل dynamo و GSMP لمحاكاة النظم المستمرة GPSS لمحاكاة النظم المتقطعة.

الغرض الاساسي لكل لغات البرمجة المتخصصة هو جعل عمل النماذج والبرمجة لنوع معين من المشاكل ابسط ما يكون واقل تعقيدا في التنفيذ طبقا لذلك يجب ان توفر لغة المحاكاة للاغراض الخاصة الاتي:

١. هياكل لعمل نماذج لمفردات النظام مثل صرافي البنك.
٢. خواص تلك المفردات مثل اوقات الانتظار.
٣. دالة لوصف تطور النظام مع الزمن.

ويجب ان يكون المستفيد قادر علي التحكم في تطور المحاكاة عبر الزمن.

برامج المحاكاة Simulation Software :

يمكن تقسيم البرامج المستخدمة في تطور المحاكاة الي ثلاث مجموعات:

١. لغات البرمجة ذات الاستخدام العام General Purpose Programming Languages تمتاز بالمرونة والبساطة وهي مناسبة لدراسة اساسيات وتقنيات محاكاة الانظمة المتقطعة مثل Fortran وC وC++ و Java .
٢. لغات برمجة المحاكاة مثل GPSS/II وSIMAN .
٣. بيئات المحاكاة Simulation Environment أو ما يعرف بلغات المحاكاة simulation languages وتمتاز بانها جيدة في سرعة بناء النماذج كما انها تحوي ميزات مبنية داخلها علاوة علي ذلك تزودنا بالرسومات والصور المتحركة وكمثال لها Arena, Automod.

تاريخ برامج المحاكاة History of Simulation Software:

نجد ان تطور برامج المحاكاة مر بست فترات وهي :

١. فترة البحث (1955-60) The Period of search.
٢. فترة الميلاد (1961-65) The Advent.
٣. فترة التكوين (1966-70) The Formative Period .
٤. فترة التوسع (1971-78) The Expansion Period.
٥. فترة التعزيز واعادة البناء The Period of Consolidation and Regeneration (1979-86).
٦. فترة البيئات المتكاملة (-1987) The Period of Integrated Environment.

اختيار برامج المحاكاة selection simulation software :

تختار برامج المحاكاة علي حسب:

-مميزات بناء نموذج المحاكاة Model building feature

١- قابلية تحليل البيانات المدخلة input data analysis capability

٢- بناء النموذج التخطيطي graphical model building

٣- التوجيه الشرطي conditional routing

٤- برمجة المحاكاة simulation programming

٥- التعبير syntax

٦- مرونة المدخلات input flexibility

٧- تبسيط النموذج modelling conciseness

٨- المكونات والقوالب المتخصصه specialized components and templates

٩- الكيانات المبنية بواسطة المستخدم user-built objects

١٠-التداخل مع لغات البرمجة العامة interface with general programming language

-بيئة وزمن التشغيل Runtime environment

١- سرعة التنفيذ execution speed

٢- حجم النموذج model size (عدد المتغيرات والعوامل)

٣- المنقح التفاعلي interactive debugger

٤- حالة واحصائية النموذج model status and statistic

-مميزات ناتج المحاكاة Output features

١- تحقيق الافضلية optimization

٢- التقرير القياسي standardized report

٣- التحليل الاحصائي statistical report

٤- العمل التخطيطي business graphic

٥- استيراد الملف (من قاعدة بيانات مثلا) file export

-دعم البائع وتوثيق المنتج Vendor support and product documentation

١- التدريب training

٢- التوثيق documentation

٣- نظام المساعدة help system

٤- البحوث tutorials

٥- الدعم support

٦- الترقية والصيانة upgrades, maintenance

- ميزات الصورالمتحركة

١- انواع الصور المتحركة type of animation

٢- استيراد ملفات الرسم والكائنات import drawing and objects file

٣- الابعاد dimension

٤- نوعية الحركة quality of motion

٥- مكتبات الكائنات العامة libraries of common objects

٦- خطوات العرض display step

٧- الحركة movement

٨- النافذة navigation

٩- المناظر views

١٠- قابلية اختيار الاجسام selectable objects

١١- العتاد المطلوب hardware requirements

امثلة للمحاكاةSimulation Examples

وتتم دراسة هذه الامثلة لاستنباط جدول محاكاة simulation table ويمكن تكوين هذا الجدول يدويا او باستخدام الحاسوب (اوراق العمل) وجدول المحاكاة هو طريقة منظمة لتعقب حالة النظام خلال فترة زمنية محددة وتوفر هذه الامثلة نظرة كافية لمنهج محاكاة النظم المتقطعة وتستخدم الاحصاء الوصفي descriptive Statistics لتأكيد اداء النظام.

وهنالك ثلاث خطوات تتبعها مع كل نظم المحاكاة:

١. تحديد شكل وومواصفات كل مدخلات المحاكاة ويمكن احيانا لتلك المدخلات ان تنتج باستخدام التوزيعات الاحتمالية اذا كان النظام مستمر او متقطع.

٢. بناء جدول المحاكاة ويختلف جدول المحاكاة من نظام لاخر ويتطور حسب نوع مشكلة.

الجدول التالي جدول (٢) يوضح جدول محاكاة :

Inputs	*Response*
	1
	2
	3
	.
	.
	.

Table(2): Simulation Table

وفي هذه المثال نجد عدد المدخلات هو P مدخل x_{ij} , j =1,2,...,p واستجابة واحدة هي y_i لكل i=1,2,...,N وننشي الجدول بعينة بيانات لتكرار ١.

٣. لكل تكرار i يجب انتاج قيمة لكل مدخل p وحساب الدالة وحساب قيمة الاستجابة y_i ويمكن استخدام التوزيعات الاحتمالية لحساب قيم المدخلات لكل خطوة وتعتمد الاستجابة الحالية علي المدخلات مع واحدة او اكثر من الاستجابات السابقة.

محاكاة انظمة الصفوف Simulation of Queuing Systems:

يوصف نظام الصفوف بعدد العناصر المكونه للنظام وطبيعة وصول تلك العناصر وميكانيكية الخدمة وسعة النظام وكيفية ضبط الصف . والشكل التالي يوضح صف بسيط مكون من قناة خدمة واحدة.

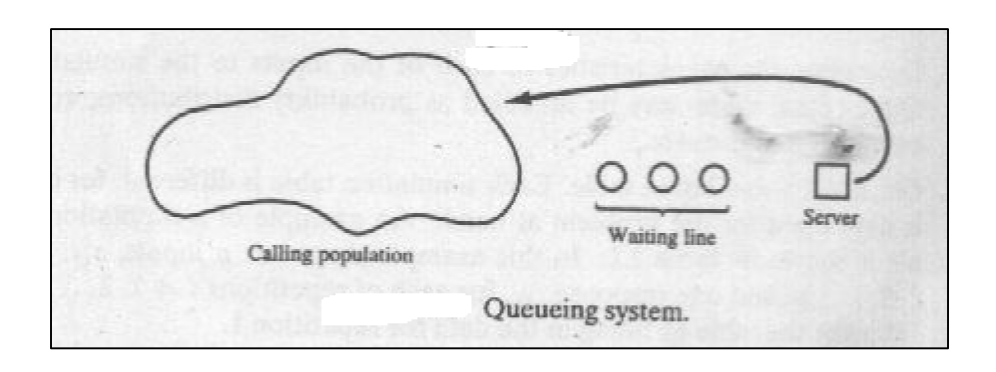


Fig.(4): Queuing system

مميزات الصف السابق :

١. عدد العناصر غير محدد، وهذا يعني اذا غادر عنصر وانضم الي صف الانتظار او انجاز الخدمة هذا لا يغير ولا يؤثر في معدل الوصول للعناصر الاخري التي تحتاج للخدمة.

٢. الوصول للخدمة يتكرر مرة واحدة خلال فترة زمنية محددة في شكل عشوائي، لان العناصر في صف الانتظار فانهم في اخر الامر سوف تنجز لهم الخدمة.

٣. اوقات الوصول لبعض العناصر لها طول عشوائي علي حسب التوزيع الاحتمالي الذي لا يتغير خلال فترة زمنية.

٤. سعة النظام ليس لها مدي محدد وهذه يعني ان اي عدد من العناصر لها المقدرة علي ان تتنظر في الصف.

٥. العناصر التي ادت الخدمة مرتبة علي حسب وصول **FIFO** باستخدام صف خدمة ذو قناة واحدة للخدمة **single server**.

نعرف اوقات الوصول والخدمات بواسطة توزيع الزمن بين اوقات الوصول واوقات الخدمة بالتتالي. لكل صف بسيط ذو قناة خدمة واحدة معدل زمن الوصول يجب ان يكون اقل من معدل زمن الخدمة في حالة نمو خط الانتظار من غير حد. ففي مثل هذه الحالة نسمي الصف صف انفجاري او غير مستقر.

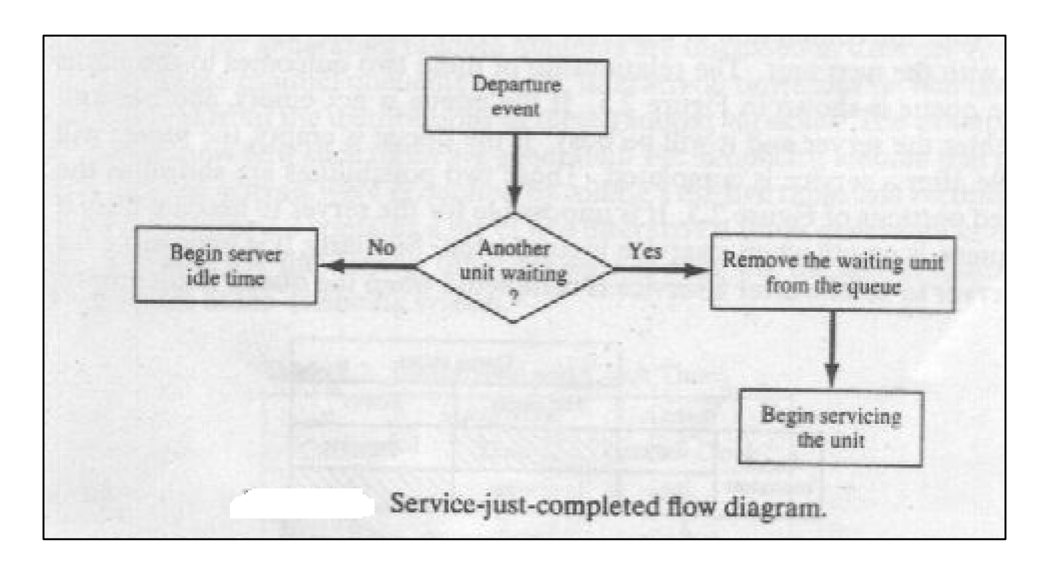
قبل تقديم محاكاة انظمة الصفوف من الضروري تغطية المفاهيم الاتية:

- **حالة النظام system state** : وهي عدد العناصر في النظام وحالة مؤدي الخدمة **server** هل هو مشغول او عاطل .
- **الاحداث events** وهي مجموعة من الحالات التي تسبب تغيير في حالة النظام وفي حالة صف الخدمة الفردية او الصف ذو قناة خدمة واحدة هنالك حدثين يؤثران في حالة النظام وهما دخول عنصر في النظام (حدث الوصول) والحدث الثاني هو اكتمال الخدمة لعنصر (حدث المغادرة). ويحتوي نظام الصف علي الخادم **sever** والوحدة التي يجب ان تقدم لها الخدمة والوحدات الاخري في الصف (المنتظرين).
- **ساعة المحاكاة Simulation clock**:

وهي تستخدم لتعقب tracking زمن المحاكاة.

الشـكل التـالي (شـكل (٥)) يوضـح اذا كـان العنصـر قـد اكمـل الخدمـة ام لا مـع

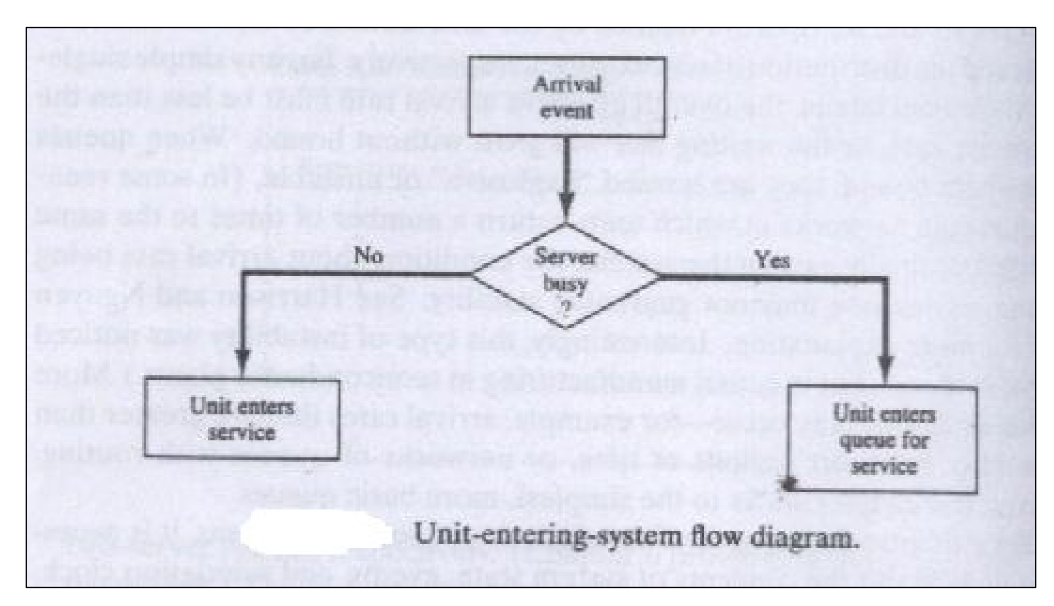
ملاحظة ان هنالك حالتان فقط لمقدم الخدمة اما مشغول **busy** او عاطل **idle** :



Service-just-completed flow diagram.

Fig.(5) Service-just-completed flow diagram

يتكـرر حـدث الوصـول عنـد دخـول عنصـر للنظـام والشـكل التـالي(شـكل (٦)) يوضـح حـدث الوصول:



Fig(6): Unit-entering-system flow diagram

العنصـر قـد يجـد مقـدم الخدمـة **server** امـا مشـغول او عاطـل ، لـذلك فـان العنصـر قـد يبـدأ بالخدمة مباشرة اذا كان مقدم الخدمة عاطل **idle** او قد يدخل الي صف الخدمة (الانتظار) اذا كان مقدم الخدمة مشغول **busy** علي حسب حالة مقدم الخدمة.

ومن المستحيل ان يكون مقدم الخدمة عاطل وصف الخدمة غير فارغ (يوجد عنصر واحد علي الاقل في صف الخدمة) كما في الشكل التالي (شكل(٧)):

		Queue status	
		Not empty	Empty
Server Status	Busy	Enter queue	Enter queue
	Idle	Impossible	Enter service

Fig.(7): Server Status

بعد انتهاء الخدمة يمكن لمقدم الخدمة ان يكون عاطل او يبقي مشغول مع العنصر التالي ، *ومن المستحيل* ان يصبح مقدم الخدمة مشغول اذا كان صف الخدمة خالي، ومن المستحيل ان يكون مقدم الخدمة عاطل بعد انتهاء الخدمة عندما يكون الصف غير خالي كما في الشكل التالي(شكل (٨)):

		Queue status	
		Not empty	Empty
Server Outcomes	Busy		Impossible
	Idle	Impossible	

Fig.(8): Server Outcomes

مثال:

اذا كان لدي احد البنوك صراف آلي واراد البنك دراسة وقت انتظار عملائه بغرض توسيع الموقع عن طريق ادخال صرافين جدد اذا ما كان متوسط زمن الانتظار طويل جدا.

خطوات الحل:

الحل اليدوي

١. الاستعانة باستشاري لملاحظة عمل النظام اذ يلاحظ هذه الشخص اوقات انتظار العملاء الذين يتعاملون مع النظام ولتكن w1,w2,w3,...,W50 لاول ٥٠-عميل، وبعد ذلك يمكن استخدام الاحصاء البسيط مثل حساب المتوسط average او استخدام الاحصاء الاكثر تعقيدا مثل منحني الرتب للنسب المئوية percentile rank curve للبيانات ومن ثم اتخاذ القرار المناسب.

تتطلب مشكلة اتخاذ القرار decision making problem تحليل احصائي بسيط مثل الوسط الحسابي w او تتطلب تحليل احصائي معقد اذا ما اخذنا متوسط زمن الانتظار كدالة في الزمن t مثل w(t)

٢. جمع البيانات: يمكن جمع البيانات $W_1,W_2,W_3,...,W_{50}$ من العالم الحقيقي ومن ثم استخدام الوسائل الرياضية والاحصائية المناسبة ثم الوصول الي قرار.

الحل باستخدام محاكاة الحاسوب:

نفس الخطوات السابقة، الاختلاف بدلا من ملاحظة وجمع البيانات في العالم الحقيقي يمكن استخدام دالة لانتاج البيانات داخل الحاسوب.

مثالك اذا كان القرار هو زيادة عدد الصرافين فالسؤال المطروح هو كم العدد المطلوب فباستخدام دالة مع ملاحظة التغيير في w مع زيادة عدد الصرافيين k يتم اختيار k الاكثر كفاءة للتكلفة. وهذ ما يسمي بالاختبار قبل التنفيذ بالمقارنه مع العالم الحقيقي يكون اجراء التجارب علي النظام مكلف جدا ومعقد جدا وفي المثال السابق لا يمكن بناء نظام يستخدم عدد متغير من الصرافيين.

تتميز محاكاة الحاسوب بان البيانات تجمع بطريقة عشوائية بينما تجمع البيانات في العالم الحقيقي بالملاحظة والتسجيل.لذلك من الضرور دراسة:

١. كيفية بناء النموذج.

٢. كيف يمكن انتاج الارقام العشوائية$W_1,W_2,W_3,...,W_{50}$ بواسطة الحاسوب.

٣. كيف يمكن استخدام مهارات المحلل لعمل افضلية للنظام optimize.

٤. كيف يمكن استخدام عملية المحلكاة في حل مشاكل رياضية واحصائية معقدة يصعب حلها بالطرق التحليلية.

يجب ملاحظة الاتي.

١. نتائج المحاكاة تكون احصائية في طبيعتها ويلزم ذلك تحليل احصائي لاستخدامها استخداما مناسبا.

٢. التكرار ضروري للوصول للاستخدام المناسب ، وغالبا تشغيل المحاكاة مرة واحدة لا يحقق الاهداف.

٣. تحتاج النماذج الي التأكد من صحتها عن طريق مقارنة النتائج (النتائج الحالية مع نتائج معروفة في النظم الموجودة.

يمكن التفكير في محاكاة النماذج كطريقة تجريبية وتطبيقية يجب ان تنجز:

١. وصف سلوك وتصرفات النظام.

٢. بناء نظريات وافتراضات تحسب السلوك المتوقع.

٣. استخدام النموذج للتبؤ بمستقبل للنظام.

تنفيـذ المحاكـاة بالحاسـوب Computer Implementation Of Simulation:

الصعوبة المصاحبة لتنفيذ نماذج المحاكاة بالحاسوب باستخدام لغات برمجة ذات اغراض عامة مثل البيسك والباسكال ادت الي اعداد العديد من لغات محاكاة ذات اغراض خاصة وتكون هذه اللغات في بعض الاحيان ذات غرض خاصة بدرجة لا تكون مناسبة لمحاكاة نظم اخري.مثل dynamo وGSMP, لمحاكاة النظم المستمرة، و GPSS لمحاكاة النظم المتقطعة.

الغرض الاساسي لكل لغات البرمجة المتخصصة هو جعل عمل النماذج والبرمجة لنوع معين من المشاكل ابسط ما يكون واقل تعقيدا في التنفيذ طبقا لذلك يجب ان توفر لغة المحاكاة للاغراض الخاصة الاتي:

٤. هياكل لعمل نماذج لمفردات النظام مثل صرافي البنك.

٥. خواص تلك المفردات مثل اوقات الانتظار.

٦. دالة لوصف تطور النظام مع الزمن.

ويجب ان يكون المستفيد قادر علي التحكم في تطور المحاكاة عبر الزمن.

مثال صف خدمة ذو قناة واحدة Single-Channel Queue :

اذا كان لدينا محل سوبر ماركت صغير ولدية محطة حساب واحدة ، يصل الزبائن لهذه المحطة في زمن عشوائي يتراوح ما بين ١ الي ٨ دقائق مع تساوي الاحتمالية لاي زمن حدوث كما في ⊫ (٣). وكان تغير زمن اداء الخدمة يتراوح ما بين ١–٦ دقائق مع احتمالية حدوث كما موضحة في جدول(٤). المطلوب دراسة وفهم النظام بالمحاكاة باستخدام ٢٠ زبون كعينه.

Random -Digit Assignment	الاحتمالية التراكميةCumulative probability	الاحتمالية Probability	الزمن بين الوصول Time between Arrival (minutes)
١٢٥_٠٠١	٠.١٢٥	٠.١٢٥	1
٢٥٠_١٢٦	٠.٢٥٠	٠.١٢٥	2
٣٧٥_٢٥١	٠.٣٧٥	٠.١٢٥	3
٥٠٠_٣٧٦	٠.٥٠٠	٠.١٢٥	4
٦٢٥_٥٠٠	٠.٦٢٥	٠.١٢٥	5
٧٥٠_٦٢٦	٠.٧٥٠	٠.١٢٥	6
٨٧٥_٧٥١	٠.٨٧٥	٠.١٢٥	7
٠٠٠_٨٧٦	١.٠٠٠	٠.١٢٥	8

Table(3): Distribution of Time Between Arrivals

Random -Digit Assignment	الاحتمالية Cumulative التراكمية probability	الاحتمالية Probability	زمن الخدمة Service Time (minutes)
١٠_٠١	٠.١٠	٠.١٠	١
٣٠_١١	٠.٣٠	٠.٢٠	٢
٦٠_٣١	٠.٦٠	٠.٣٠	٣

٨٥_٦١	٠.٨٥	٠.٢٥	٤
٩٥_٨٦	٠.٩٥	٠.١٠	٥
٠٠_٩٦	١.٠٠	٠.٠٥	٦

Table(4):Distribution of Service Time

باستخدام بعض الخواص الاحصائية يمكننا تحديد بعض الخصائص لاحتمال زمن وصول الزبائن ومن هذه الخصائص حساب الاحتمالية التراكمية **Cumulative probability** وتحديد مدي هذه الاحتمالية **Random -Digit Assignment** كما موضح باخر عمودين في جدول(٤).

الحل:

اولا: نجد ان استخدام ٢٠ زبون كعينة يعتبر عدد قليل ولكن نجده عدد مناسب كوسيلة شرح وفهم لبناء نظام باستخدام المحاكاة.

ثانيا: نحتاج الي مجموعة من الارقام العشوائية لانتاج زمن الوصول الي محطة الحساب وزمن الخدمة. ويمكن انتاج الارقام العشوائية باستخدام الدوال المبنية في الحاسب وهي تنتج ارقام عشوائية ما بين ٠ و ١ ، او استخدام الارقام العشوائية اليدوية وهي نجدها كملاحق في معظم الكتب. وفي هذه المثال سوف تستخدم الارقام العشوائية اليدوية (مرفق جدول به ارقام عشوائية). نلاحظ الي اخر عمود في كل جدول لتحديد طبيعة الارقام العشوائية المستخدمة في الجدول الاول نستخدم ارقام عشوائية مكونه من ثلاث خانات وفي الجدول الثاني نستخدم ارقام مكونة من خانتين علي حسب طبيعة الارقام في كل عمود مع ملاحظة عدم تكرار استخدام نفس الرقم العشوائي.

ثالثاً : تحديد زمن الوصول وزمن الخدمة لكل زبون علي حسب الرقم العشوائي المختار مثلا في حالة زمن الوصول اذا تم اختيار الرقم ٩١٣ نجده يقع ما بين ٨٧٠_٠٠٠ وهذا يعني ان زمن وصول هذه المستفيد هو ٨. ونكرر هذه الخطوة الي ان تتم تعبئة زمن الوصول لكل ٢٠ زبون.

وبنفس الطريق تتم تعبة زمن الخدمة لل ٢٠ زبون كما موضح في جدول**(5)** = **(6)**.

Customer	*Random Digits*	*Tune between Arrivals (Minutes)*	*Customer*	*Random Digits*	*Time between Arrivals (Minutes)*
1	—	-	11	109	1
2	913	8	12	093	1
3	727	6	13	607	5
4	015	1	14	738	6

5	948	8	15	359	3
6	309	3	16	888	8
7	922	8	17	106	1
8	753	7	18	212	2
9	235	2	19	493	4
10	302	3	20	535	5

Table(5):Time between Arrivals determination

Customer	*Random Digits*	*Service Time (Minutes)*	Customer	*Random Digits*	*Service Time (Minutes)*
I	84	4	11	32	3
2	10	1	12	94	5
3	74	4	13	79	4
4	53	3	14	05	1
5	17	2	15	79	5
6	79	4	16	84	4
7	91	5	17	52	3
8	67	4	18	55	3
9	89	5	19	30	2
10	38	3	20	50	3

Table(6):Service Time Generated

رابعا : بناء جدول المحاكاة:

وجوهر المحاكاة هو بناء جدول المحاكاة، ونجد ان تصميم هذه الجدوال يكون مهمة محلل النظام وتصمم تلك الجدول للمشكلة المطروحة مع اضافة بعض الاعمدة للاجابة علي الاسئلة المطروحة.

الخطوة الاولي هي تعبئة الخلايا للزبون الاول:

- زمن الوصول للزبون الاول هو ٠ (صفر).
- تبدأ الخدمة مباشرة وتنتهي عند الزمن ٤.
- استغرق الزبون الاول ٤ دقائق.

الزبون الثاني:

- وصل في الزمن ٨.
- لذلك الخدمة تعطلت ٤ دقائق

الزبون الرابع:

- وصل هذا البون في الزمن ١٥.
- الخدمة لم تبدا له الا في الزمن ١٨.
- اذا انتظر هذا الزبون ٣ دقائق.

الجـــدول التـــالي ((٧)) يوضـــح جـــدول المحاكـــاة لـــ ٢٠ زبـــون.

10. Simulation Table for Queueing Problem

A tomer	B Time Since Last Arrival (Minutes)	C Arrival Time	E Time Service Begins	F Time Customer Waits in Queue (Minutes)	G Time Service Ends	H Time Customer Spends in System (Minutes)	I Idle Time of Server (Minutes)
1	–	0	0	0	4	4	0
2	8	8	8	0	9	1	4
3	6	14	14	0	18	4	5
4	1	15	18	3	21	6	0
5	8	23	23	0	25	2	2
6	3	26	26	0	30	4	1
7	8	34	34	0	39	5	4
8	7	41	41	0	45	4	2
9	2	43	45	2	50	7	0
10	3	46	50	4	53	7	0
11	1	47	53	6	56	9	0
12	1	48	56	8	61	13	0
13	5	53	61	8	65	12	0
14	6	59	65	6	66	7	0
15	3	62	66	4	71	9	0
16	8	70	71	1	75	5	0
17	1	71	75	4	78	7	0
18	2	73	78	5	81	8	0
19	4	77	81	4	83	6	0
20	5	82	83	1	86	4	0
				56		124	18

Table(7): Simulation Table for Queuing Problem

خامساً**:** الحسابات واستخلاص النتائج:

وفـي هـذه الخطـوة تحسـب بعـض الاحصـلئيات لتحديـد كفـاءة النظـام وتختلـف الاحصـائيات المستخدمة علي حسب طبيعة كل نظام.

1. The average waiting time for a customer is 2.8 minutes. This is determined in the following manner:

$$\text{average waiting time (minutes)} = \frac{\text{total time customers wait in queue (minutes)}}{\text{total numbers of customers}}$$

$$= \frac{56}{20} = 2.8 \text{ minutes}$$

2. The probability that a customer has to wait in the queue is 0.65. This is determined in the following manner:

$$\text{Probability (wait)} = \frac{\text{number of customers who wait}}{\text{total number of customers}}$$

$$= \frac{13}{20} = 0.65$$

3. The fraction of idle time of the server is 0.21. This is determined in the following manner:

$$\frac{\text{total idle time of server (minutes)}}{\text{total run time of simulation (minutes)}} = \text{Probability of idle Server}$$

$$= \frac{18}{86} = 0.21$$

The probability of the server being busy is the complement of 0.21, or 0.79.

4. The average service time is 3.4 minutes, determined as follows:

$$\frac{\text{total service time (minutes)}}{\text{total number of customers}} = \text{Average service time (minutes)}$$

$$= \frac{68}{20} = 3.4 \text{ minutes}$$

This result can be compared with the expected service time by finding the mean of the service-time distribution using the equation

$$E(S) = \sum_{s=0}^{\infty} sp(s)$$

Applying the expected-value equation to the distribution in Table (2) gives an expected service time of:

$$= 1(0.10) + 2(6.20) + 3(0.30) + 4(0.25) + 5(0.10) + 6(0.05)$$
$$= 3.2 \text{ minutes}$$

The expected service time is slightly lower than the average service time in the simulation. The longer the simulation, the closer the average will be to *E(S).*

5. The average time between arrivals is 4.3 minutes. This is determined in the following manner:

between arrivals average time between = sum of all times (minutes)

number of arrivals -1 arrivals (minutes)

$$= \frac{82}{19} = 4.3 \text{ minutes}$$

One is subtracted from the denominator because the first arrival is assumed to occur at time 0. This result can be compared to the expected time between arrivals by finding the mean of the discrete uniform distribution whose endpoints are $a = 1$ and $b = 8$. The mean is given by:

$$E(A) = \frac{a+b}{2} = \frac{1+8}{2} = 4.5 \text{ minutes}$$

The expected time between arrivals is slightly higher than the average however, as the simulation becomes longer, the average value of the time between arrivals will approach the theoretical mean, E (A).

6. The average waiting time of those who wait is 4.3 minutes. This is mined in the following manner:−deter

total time customers wait in queue (minutes) =Average waiting time of

total number of customers who those who wait (minutes)

wait

$$= \frac{56}{13} = 4.3 \text{ minutes}$$

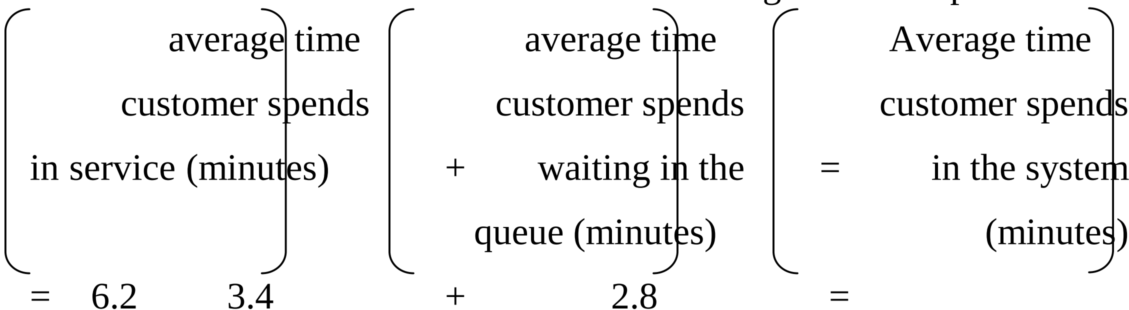
7. The average time a customer spends in the system is 6.2 minutes. This can be determined in two ways. First, the computation can be achieved by the following relationship:

average time customer = tota1 time customers spend in the
total number of customers system(minutes) spends in the system (minutes)

$$= \frac{124}{} = 6.2 \text{ minutes}$$

20

The second way of computing this same result is to realize that the following relationship must hold:



= 6.2 3.4 + 2.8 =

minutes

النتائج:

- معظم الزبائن ينتظرون ، رغم من ان متوسط زمن الانتظار غير زائد.
- مقدم الخدمة ليس له زمن تعطيل كبير.
- الهدف من النتائج يعتمد علي موازنة تكلفة زمن الانتظار مع تكلفة زيادة مقدمي الخدمة.

المفاهيم الأساسية لمحاكاة النظم المتقطعة

General Principle to Concept In Discrete-Event Simulation

في هذا الجزء نقدم ونشرح المفاهيم الاساسية المستخدمة في اغلب حزم محاكاة النظم المتقطعة وهي لا ترتبط بحزمة معينه.

مفاهيم في محاكاة النظم المتقطعة **Concepts in discrete-event simulation:**

٦. النظام **system**: مجموعة من الكائنات ترتبط مع بعضها البعض بصورة ما لتحقيق عدد من الاهداف.

٧. النموذج **model** : هو وصف مبسط للنظام.

٨. حالة النظام **system state** هي مجموعة من المتغيرات الضرورية لوصف النظام في اي وقت بالنسبة لاهداف النظام.

٩. الكيان **Entity** وهو اهم كائن في النظام.

١٠. الصفات **Attribute** وهي خصائص الكيان ومكوناته.

٨. القائمة **List** هي مجموعة من الكيانات مرتبة بطريقة منطقية.

٩. الحدث **Event** وهو حالة تُحدث تغبير فوري يغير حالة النظام.

١٠. ملاحظة الحدث **Event Notice** وهي سجل للحدث ويحتوي عادة علي اسم الحدث وفترته الزمنية (event type, event time).

١١. قائمة الاحداث **Event List** وهي قائمة بالاحداث المستقبلية والاحداث الوشيكة الحدوث وترتب علي حسب وقوعها وتكتب في شكل قائمة تسمي قائمة الاحداث المستقبلية(Future Event List (FEL.

١٢. النشاط **Activity** وهي الفترة الزمنية لانجاز عمل معين. وهي مدة من الزمن محددة الطول وتعرف منذ البداية مثل زمن الوصول وزمن الخدمة.

١٣. التأخير **Delay** وهي فترة غير محدة الطول .

١٤. الساعة **Clock** وهي متغير يمثل زمن المحاكاة.

- تستخدم بعض حزم المحاكاة مصطلحات مختلفة لنفس المفاهيم السابقة. مثلا القائمة List تسمي في بعض الحزم مجموعة Set او صف Queue او سلسلة chain.
- ترتيب الكيانات في قائمة يُرتب دائما بواسطة قاعدة مثل: First In First Out (FIFO) او Last In First Out (LIFO) او ترتب بصفة من الكيان مثل الاولوية او التاريخ
- قائمة الاحداث المستقبلية FEL تُرتب دائما علي حسب زمن الحدث event Time المسجل في ملاحظة الحدث Event Notice.

- والنشاط Activity يمثل زمن الخدمة، زمن الوصول او اي معالجة زمنية اخري لها مده زمنية معرفة بواسطة واضع النموذج modeler. ومدة النشاط يمكن ان تحدد بعدة طرق ويمكن ان تكون:

٤. محددة deterministic مثلا تكون دائما ٥ دقائق.

٥. احصائية statistical مثال رقم عشوائي من باحتمالية متساوية.

٦. دالة تعتمد علي متغيرات النظام (او/و) صفات الكيان مثال زمن التحميل لسفينة كدالة لحمولة السفينة ومعدل التحميل كطن في الساعة.

- التأخير delay يسمي في بعض الاحيان بالانتظار الشرطي conditional wait والنشاط يسمي الانتظار غير الشرطيunconditional wait. اكتمال النشاط activity هو حدث event. وغالباً يسمي الحدث الابتدائي primary event ويدار باحلال ملاحظة الحدث event notice في FEL.
- وعلي العكس نجد ان التأخير يدار بواسطة احلال الكيان المساعد associated entity في قائمة اخري او في صف الانتظار. واكتمال التأخير يسمي في بعض الاحيان شرطي conditional او حدث ثانوي secondary event ولكن هذه الاحداث لا تمثل بواسطة event notice ولا تظهر في FEL.
- والانظمة التي تُعالج هنا هي انظمة متحركة وهذه يعني انها تتتغير خلال فترة زمنية لذلك حالة النظام، وصفات الكيان وعدد من الكيانات النشطة والنشاطات والتأخير تتغير جميعها خلاة فترة من الزمن.
- والزمن نفسه يمثل بواسطة متغير يسمي clock.

مثـال: تطبيـق تلـك المفـاهيم علـي صـف خدمـة ذو مقـدم خدمـة واحـد: Single Channel Queue

نجـد ان هـذا النظـام يتكـون مـن الزبـائن الـذين ينظـرون فـي الصـف مـن اجـل الحسـاب بالاضـافة الـي مقدم خدمـة واحد one check point. لذلك نجد ان لهذا النموذج المفاهيم التالية:

١) حالة النظام **system State** : وهنالك حالتان للنظام وهما:

- **LQ (t) represents number of customers in waiting line.**
- **LS (t) represents number of served (0, 1) at time t.**

٢) الكيانات **entities**: وهي

- مقدم الخدمة **server**.
- الزبائن او العملاء **customer**.
- حالة المتغيرات اعلاه.

٣) الاحداث **events**: وهي

- **Arrival (A).**
- **Departure (D).**
- **Stopping event (E), schedule to occur at time 60.**

٤) ملاحظة الحدث **Event notice**: وهي

- **(A, t) Representing an arrival event to occur at future time t.**
- **(D, t) Representing departure event to occur at future time t.**
- **(E,60) Representing the simulation-stop event at future time 60.**

٥) النشاطات **Activities**: وهي

- **Inter arrival time (table 3).**
- **Service time (table 4).**

٦) التأخير **Delay**: وهو زمن الذي يقضيه الزبون في صف الانتظار.

تعريف مكونات ومفاهيم النموذج تجهز وصف ثابت للنموذج بالاضافة الي ان وصف العلاقات المتغيرة والتفاعلات بين المكونات نحتاج اليه.

وهنالك بعض الاسئلة تحتاج الي اجابة وهي:

- كيف ان كل حدث يؤثر في حالة النظام **system state** وصفات الكيان**entity attribute** ومجموعة المكونات؟
- كيف تعرف النشاطات **activities**؟ هل هي محددة ام احتمالية ام معادلات رياضية اخري؟ وكيف ان الحدث يحدد بداية ونهاية كل نشاط؟ وهل يمكن للنشاط ان يبداء بصرف النظر عن حالة النظام مثلا في حالة الالت النشاط لايمكن ان يبداء حتي تكون الالة عاملة وغير معطلة وليست تحت الصيانة.
- اي من الاحداث تحدد البداية والنهاية لكل نوع تأخير **delay**؟
- ما هي حالة النظام في الزمن ٠ **(time 0)** ؟ او ماهي الاحداث التي يجب ان تنجز في الزمن ٠ ليبداء النموذج او التي تجعل المحاكاة تبداء؟

والصورة (اللقطة) لمحاكاة هي حالة النظام في وقت محدد **clock=t** وهي تحتوي علي:

- ✓ حالة النظام في الفترة **t** .
- ✓ قائمة **FEL** لكل النشاطات الحالية وزمن نهاية كل نشاط من تلك النشاطات.
- ✓ الحالة لكل الكيانات وكل الاعضاء الحاليين لكل المجموعات.

✓ القيم االحالية للاحصائيلت التراكمة والعدادات التي تستخدم لحساب ملخص الاحصائيات في نهاية المحاكاة.

توجد بعض الحالات او الصور او اللقطات في الشكل التالي:

Clock	System State	Entities and Attributes	Set 1	Set 2	...	Future Event List, FEL	Cumulative Statistics and Counters
t	$(x, y, z, \ldots)$					$(3, t_1)$ – Type 3 event to occur at time t_1 $(1, t_2)$ – Type 1 event to occur at time t_2	
	. . .						

Figure Prototype system snapshot at simulation time t.

Fig.(9): Prototype system snapshot at simulation time t

جدولة الأحداث/خوارزمية زيادة الزمن

The Event-Scheduling/Time advance algorithm

الآلية لزيادة زمن المحاكاة وضمان ان كل الاحداث تحدث في ترتيب تاريخي هي مؤسسة علي قائمة الاحداث المستقبلية **Future Event List (FEL)** وهذه القائمة تحتوي علي كل ملاحظات الحدث **event notices** لكل الاحداث التي جدولت لتحدث في زمن مستقبلي. وجدولة الاحداث المستقبلية تعني ان في لحظة بداية النشاط ومدته التي تحسب من توزيع احصائي ونهاية النشاط جميعهم مع زمن حدوثه توضع في **FEL**.

في اي زمن معطي **t** القائمة المستقبلية **FEL**تحتوي على الاحداث المستقبلية المجدولة سابقا وزمن حدوثهم **($t_1,t_2,..$)**. وهي مرتبة بواسطة زمن الحدوث **event time** اي ان الاحداث مرتبة تاريخيا اي ان **$t<t1<=t2<=t3<=...<=t_n$**. حيث ان:

t هي القيمة الحالية للـ **clock** لزمن المحاكاة، و **t1** هي زمن الحدث الوشيك الحدوث اي انه الحدث التالي بعد اخذ حالة النظام في الزمن **clock=t**. ويتم تغير النظام والـ **clock** تتغيير الي **$clock=t_2$** ويحذف الحدث الحالي من **FEL** وينفذ الحدث الوشيك الحدوث.

تنفيذ الحدث الوشيك الحدوث تعني ان هناك حالة جديدة للنظام في الزمن **t_1** تنشأ مؤسسة علي الحالة السابقة في الزمن **t** وطبيعة الحدث الوشيك الحدوث.وتتكرر هذه المعالجة حتي تنتهي المحاكاة.

والشكل التالي (شكل(١٠)) يوضح The Event-Scheduling/Time advance algorithm

Old system snapshot at time t

CLOCK	System State	...	Future Event List	...
t	(5, 1, 6)		$(3, t_1)$ – Type 3 event to occur at time t_1	
			$(1, t_2)$ – Type 1 event to occur at time t_2	
			$(1, t_3)$ – Type 1 event to occur at time t_3	
			.	
			.	
			.	
			$(2, t_n)$ – Type 2 event to occur at time t_n	

Event-scheduling/time-advance algorithm

Step 1. Remove the event notice for the imminent event (event 3, time t_1) from FEL

Step 2. Advance CLOCK to imminent event time (i.e., advance CLOCK from t to t_1).

Step 3. Execute imminent event: update system state, change entity attributes, and set membership as needed.

Step 4. Generate future events (if necessary) and place their event notices on FEL ranked by event time. (*Example:* Event 4 to occur at time t^*, where $t_2 < t^* < t_3$.)

Step 5. Update cumulative statistics and counters.

New system snapshot at time t_1

CLOCK	System State	...	Future Event List	...
t_1	(5, 1, 5)		$(1, t_2)$ – Type 1 event to occur at time t_2	
			$(4, t^*)$ – Type 4 event to occur at time t^*	
			$(1, t_3)$ – Type 1 event to occur at time t_3	
			.	
			.	
			.	
			$(2, t_n)$ – Type 2 event to occur at time t_n	

Figure Advancing simulation time and updating system image.

Fig.(10): Advancing simulation time and updating system image

طول ومحتويات **FEL** تتغير بتطور المحاكاة وادارة القائمة تسمي list processing

والعمليات الاساسية التي تنجز في list processing علي FEL هي:

١. ازالة الحدث الوشيك الحدوث.

٢. اضافة حدث جديد للقائمة.

٣. ازالة بعض الاحداث وتسمي الغاء الحدث.

الازالة تكون للحدث الوشيك ويكون في مقدمة القائمة، اما اضافة حدث جديد تحتاج الي بحث القائمة وعلي حسب خوارزمية البحث المستخدمة تحدد كفاءة التشغيل وهناك عدة تقنيات مستخدمة في البحث. وازالة واضافة الاحداث من FEL موضحة في شكل (١٠).

مثال: تطبيق FEL علي صف خدمة ذو مقدم خدمة واحد: Single Channel Queue

ملاحظات :

- في النموذج FEL تحتوي علي اما ملاحظتين او ثلاثة للاحداث **2 or 3 event notice**.
- تأثير احداث الوصول والمغادرة وصفت سابقا في شكل (٥) وشكل (٦) والأشكال التالية شكل (١١) وشكل (١٢) بها بعض التفصيل .

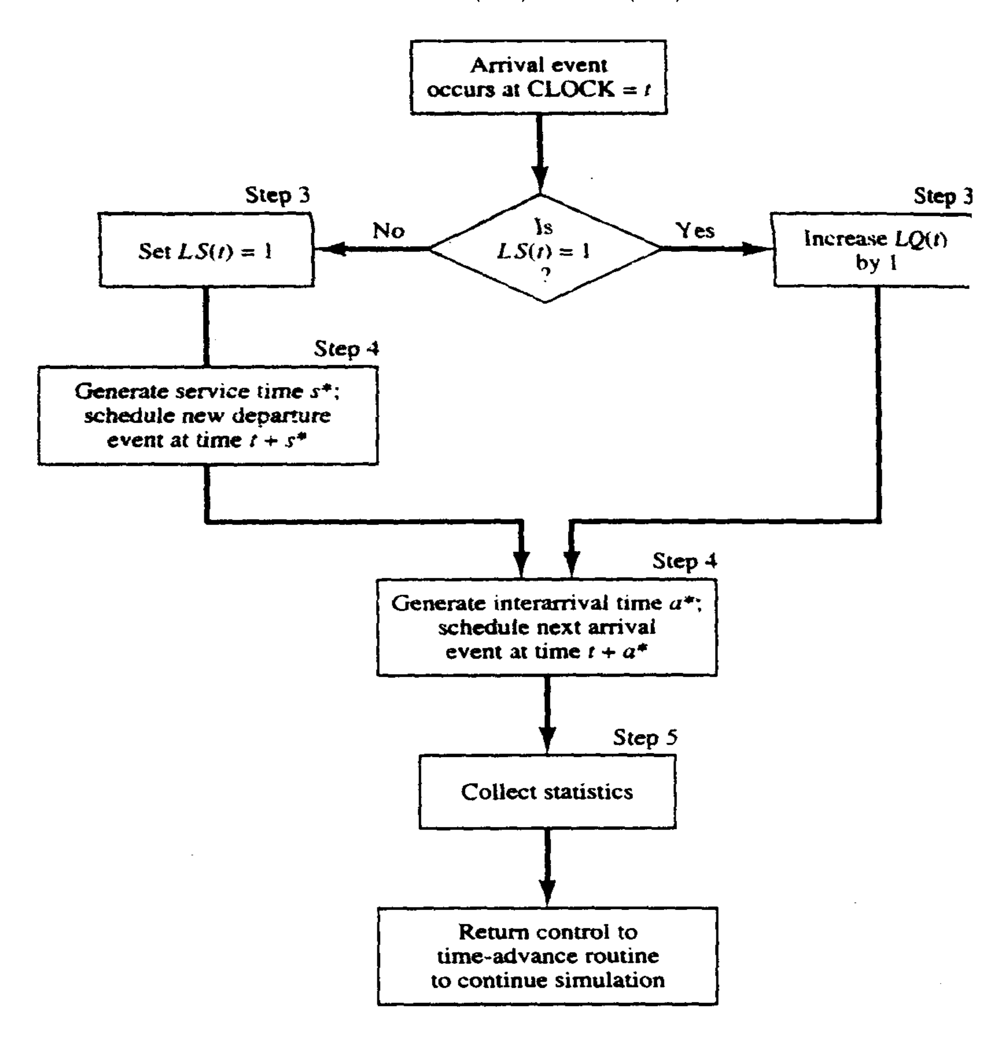


Fig.(11). Execution of the arrival event.

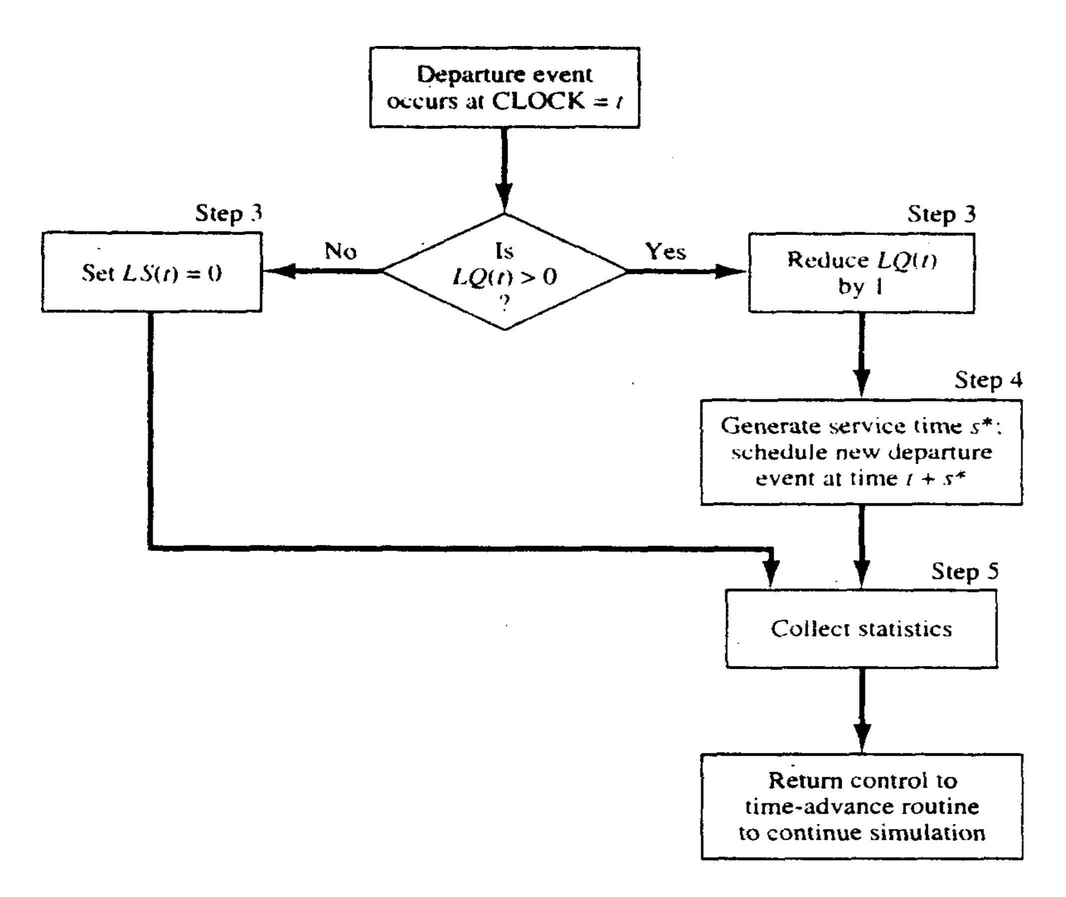


Fig.(12). Execution of the departure event

❖ جدول المحاكاة لمحطة الخدمة يوضح في الجدول التالي جدول (٩) .

Table 3. Simulation Table for Checkout Counter (Example 3.3)

Clock	System State LQ(t)	LS(t)	Future Event List	Comment	Cumulative Statistics B	MQ
0	0	1	(D, 4) (A, 8) (E, 60)	First A occurs ($a^* = 8$) Schedule next A ($s^* = 4$) Schedule first D	0	0
4	0	0	(A, 8) (E, 60)	First D occurs: (D, 4)	4	0
8	0	1	(D, 9) (A, 14) (E, 60)	Second A occurs: (A, 8) ($a^* = 6$) Schedule next A ($s^* = 1$) Schedule next D	4	0
9	0	0	(A, 14) (E, 60)	Second D occurs: (D, 9)	5	0
14	0	1	(A, 15) (D, 18) (E, 60)	Third A occurs: (A, 14) ($s^* = 4$) Schedule next D	5	0
15	1	1	(D, 18) (A, 23) (E, 60)	Fourth A occurs: (A, 15) (Customer delayed)	6	1
18	0	1	(D, 21) (A, 23) (E, 60)	Third D occurs: (D, 18) ($s^* = 3$) Schedule next D	9	1
21	0	0	(A, 23) (E, 60)	Fourth D occurs: (D, 21)	12	1

Table(9): Simulation table for checkout counter

❖ (٩) لمحطة الخدمة **checkout counter** وعلي المستخدم تغطية كل حالات النظام وتكون البداية من الحالة الاولي ويحاول بناء الحالة الثانية من السابقة كما موضح في الاشكال السابق.

❖ زمن الوصول وزمن الخدمة كما في جدول(٣) ١ :(٤) في المثال السابق وهي باختصار:

Inter arrivals	8	6	1	8	3	8	…
Service time	4	1	4	3	2	4	…

<u>مناقشة وشرح جدول المحاكاة جدول(٩)</u>

❖ الشروط الابتدائية هي وصول العميل الاول في الزمن ٠ وبدء الخدمة له وهذا يعكس في الجدول (٩) بواسطة حالة النظام **system snapshot** في الزمن صفر

(clock=0) مـع **LQ(0)=0** و **LS(0)=1** وكـل مـن حـدث الوصـول والمغـادرة موجودان في **FEL** لجدول المحاكاة وتتوقف المحاكاة في الزمن ٦٠.

❖ هناك احصائيات جديدة وهي

- **Server utilization= total server busy time (B) / total time (T_E).**
- **Maximum queue length MQ will be accumulated as the simulation progresses.**

ونجد ان **B , MQ** تحسب تراكميا مع تطور ونمو المحاكاة.

❖ العمود **Comment** اضيف لمساعدة المستخدم.

❖ **s*** و **a*** تنتج اوقات الخدمة والوصول.

❖ عند اكتمال حالة النظام في الزمن **clock=0** تبداء المحاكاة مباشرة.

❖ الحـدث الوشـيك الحـدوث هـو **(D,4)** لـذلك المتغيـر **clock** يتغيـر الـي الـزمن ٤. والحدث **(D,4)** يحذف من **FEL**.

❖ عنـد **LS(t)=4** لـ **0<=t<=4** (اي ان مقدم الخدمـة مشـغول لمـدة ٤ دقـائق) يزيـد يراكم الانشغال **busy** من **B=0** الي **B=4**.

❖ الان **FEL** بها حدثين منفصلين هما **(A,8)** و **(E,60)** وتذهب **clock** الي الزمن التالي وهو ٨ وينفذ حدث الوصول.

❖ يغطي الجدول السابق الفترة الزمنية **[0,21]** ي زمن لمحاكاة ٢١ نجد ان النظام خالي ولكن هنالك زمن وصول وشيك في الزمن ٢٣.

❖ مقدم الخدمة مشغول ل[١٢ وحدة زمنية من اصل ٢١ وحدة زمنية .

❖ اقصي طول للصف هو (١).

نجد ان هذه المحاكاة قصيرة جدا حتي يتم وضع ملاحظات واقعية.

تمرين:

مواصلة منهج المحاكاة كالسابق حتي تكتمل المحاكاة عند الزمن ٦٠ ومقارنه النتائج

التي تحصل عليها مع النتائج الموجودة سابقا.

برامج المحاكاة Simulation Software :

يمكن تقسيم البرامج المستخدمة في تطور المحاكاة الي ثلاث مجموعات:

٤. لغات البرمجة ذات الاستخام العام **General Purpose Programming Languages**. مثل **Fortran** و **C** و **C++**.

٥. لغات برمجة المحاكاة مثل **GPSS/II** و **SIMANV** .

٦. بيئات المحاكاة **Simulation Environment**.

تاريخ برامج المحاكاة History of Simulation Software:

نجد ان تطر برامج المحاكاة مر بست فترات وهي :

٧. فترة البحث **The Period of search (1955-60)**.

٨. فترة الميلاد **The Advent (1961-65)**.

٩. فترة التكوين **The Formative Period (1966-70)** .

١٠. فترة التقسع **The Expansion Period (1971-78)**.

١١. فترة – واعادة البناء **The Period of Consolidation and Regeneration (1979-86)**.

١٢. فترة البيئات المتكاملة **The Period of Integrated Environment (1987-)**.

أهداف النمذجة والمحاكاة Goal of Modelling and Simulation:

– النموذج ياخذ مجموعة من expressed assumption:

رياضية ، منطقية

علاقات رمزية بين الكيانات